MICROCOMPUTER DATABOOK 1984

OKI SEMICONDUCTOR

PREFACE

OKI has produced various types of CPU, MPU and Peripheral devices using CMOS technology, such as the MSM 80C85ARS which was released last year.

It is our pleasure to announce that the OKI Micro-computer Data Book was recently completed and released to the public.

The LSI device family described in this book are very highly reliable products that use advanced LSI technology.

This data book covers most of the OKI microcomputer series for convenient reference.

CONTENTS

OKI semiconductor

MSM5840HRS

CMOS SINGLE COMPONENT MICROCOMPUTER

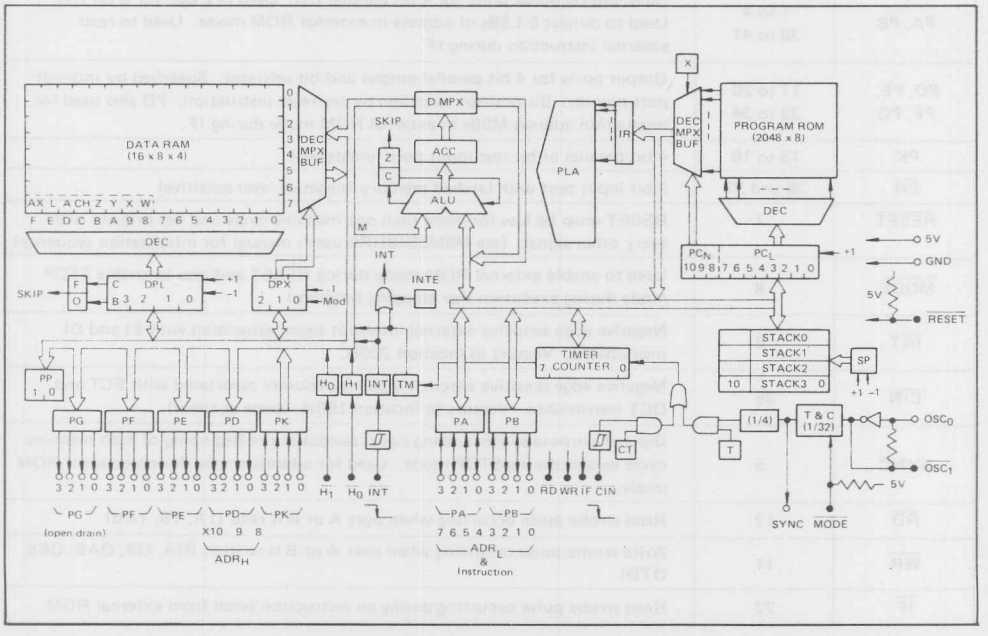
GENERAL DESCRIPTION

The OKI MSM5840HRS microcomputer is a low-power, high-performance single-chip device implemented in Complementary Metal Oxide Semiconductor technology. Integrated within this one chip are 16K bits of mask program ROM, 512 bits of data RAM, 30 Input/Output lines, a programmable timer/counter, and oscillator. Program memory is byte wide and data paths are organized in 4 bit nibbles. RAM and I/O lines are bit addressable. Up to 4K of external ROM interfaces to the 8 bit bidirectional bus. 98 instructions include binary. BCD, logical operations; bit set, reset, test 8 bit I/O; relative jumps; multifunctional instructions (increment, modify, skip); 8 bit wide table output; subroutine call and return. 94% of instructions are single byte, single cycle operations.

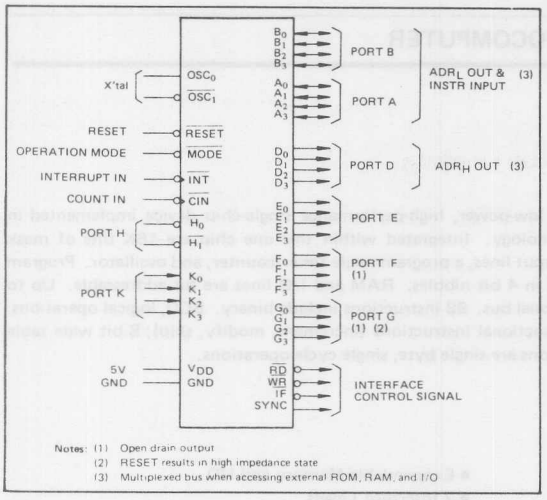
FEATURES

- Low Power Consumption – 8mW Typical
- 100% Static Logic – 100μW Standby
- 2K x 8 Internal ROM
- Up to 4K x 8 External ROM
- 128 x 4 Internal RAM
- 30 I/O Lines Incl. 8 Bit Data Bus
- Programmable 8 Bit Timer/Counter
- Self-contained Oscillator
- 98 Instructions
- Exapandable Memory and I/O
- 2 Interrupt Levels
- 4 Stack Levels
- –20° to +70°C Operating Temperature (–40° to +85°C Optional)
- 3V to 6V Operating V_{DD}
- Battery Powered or Battery Backup
- TTL Compatible (with pullups)
- 7.6μS Cycle Time @4.2MHz

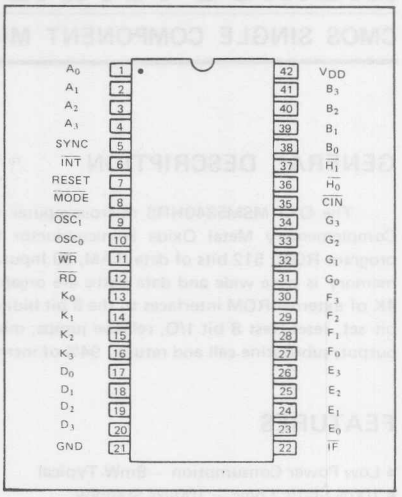
FUNCTIONAL BLOCK DIAGRAM



LOGIC SYMBOL



PIN CONFIGURATION



PIN DESCRIPTION

Designation	Pin No.	Function
GND	21	Circuit GND potential
V_{DD}	42	Main power source (+5V)
OSC_0	10	Crystal OSC input, external clock input
$\overline{OSC_1}$	9	Crystal OSC input, external clock output (not TTL compatible)
PA, PB	1 to 4 38 to 41	Quasi-bidirectional ports for 4 bit parallel I/O. Used as a pair for 8 bit I/O. Used to output 8 LSBs of address in external ROM mode. Used to read external instruction during IF.
PD, PE, PF, PG	17 to 20 23 to 34	Output ports for 4 bit parallel output and bit set/reset. Specified by internal port pointer. Bit position specified by set/reset instruction. PD also used for instruction address MSBs in external ROM mode during IF.
PK	13 to 16	4 bit parallel or bit test input port (unlatched)
$\overline{PH}$	36 and 37	2 bit input port with latched memory (negative level sensitive)
$\overline{RESET}$	7	RESET must be low for more than one machine cycle and has priority over every other signal. (see MSM5840HRS user's manual for intialization sequence)
$\overline{MODE}$	8	Used to enable external ROM mode during RESET and also to enable STOP mode during execution (for stepping program)
$\overline{INT}$	6	Negative edge sensitive external interrupt signal associated with EI and DI instructions. Vectors to location 200H.
$\overline{CIN}$	35	Negative edge sensitive external input for counter associated with ECT and DCT instructions. Vectors to location 100H. (same as timer)
SYNC	5	General purpose synchronizing signal output at the beginning of each machine cycle except during STOP mode. Used for address strobe during external ROM mode.
$\overline{RD}$	12	Read strobe pulse occurring when port A or B is read (1A, 1B, 1AB)
$\overline{WR}$	11	Write strobe pulse occurring when port A or B is written (OA, OB, OAB, OBS, OTD)
$\overline{IF}$	22	Read strobe pulse occurring during an instruction fetch from external ROM.

FUNCTIONAL DESCRIPTION

A block diagram of the MSM5840HRS is given on page 1. Each block of logic will be briefly discussed. For more information, refer to the MSM5840HRS user's manual.

Program ROM

The MSM5840HRS will address up to 4K bytes of program ROM and can have 2K bytes of internal masked ROM or all ROM may be located externally. External EPROM may be used for program development with conversion to internal ROM occurring after program debug and system checkout and verification. All instructions are byte wide. Three of the 98 instructions require two bytes of program code. The instructions are routed to a programmed logic array which generates the necessary internal control signals.

Data RAM

Data is organized in 4 bit nibbles. Internal data RAM consists of 128 nibbles, eight nibbles of which are dedicated registers accessible directly under program control. These are the general purpose registers, W, X, Y and Z, and the 4 save (exchange) registers, CH, A, L, and AX. All other DATA RAM must be addressed indirectly through the DP (data pointer) register, a seven bit pointer (directly accessible by numerous instructions) consisting of 4 bit DPL register and a 3 bit DP_H register. Any nibble of internal data RAM can be accessed through the DP register. Some Instructions, automatically change the contents of the DP register allowing efficient array processing.

Input/Output Ports

PA, PB – These two ports are quasi-bidirectional ports which can be used as simple I/O lines or used as either 4 bit or 8 bit parallel bus. Instruction fetches from external ROM occur through these ports by outputting the 8 low order bits of address during SYNC followed by an IF (instruction fetch) cycle. In addition, synchronized data transfers are possible through these prots with the I/O pin signals $\overline{RD}$ and $\overline{WR}$ associated with certain input/output instructions dedicated to these ports. In short, PA and PB can be used as a multiplexed address/instruction/data bus.

PD, PE, PF, PG – These four output ports are addressed indirectly through the TWO BIT port pointer whose contents are changed through certain instructions. These ports are bit (set/reset) addressable. PD is also used for the high order bits of address during an external instruction fetch. PF and PG are open drain outputs and PG is set high by a hardware RESET.

PK is an input port without memory, addressable either as a nibble or bit level input.

PH is a two bit input port with memory, which can be tested and reset under program control.

External Interrupt

The $\overline{INT}$ pin can be tested under program control or enabled to cause a vectored interrupt to location 200H. It is negative edge sensitive.

Timer/Counter

The timer/counter is an eight-bit counter whose input is selected under program control to be either an external signal ($\overline{CIN}$) or an internal square wave of 1/128 the frequency of the OSC_0 input (2MHz/128 = 15.625 kHz). The timer/counter can be enabled or disabled under program control as can be the associated internal interrupt which vectors to location 100H and has higher priority than the external interrupt.

Stack

The stack is a LIFO queue for storing return-from-interrupt and return-from-subroutine address information. It is eleven bits wide and 4 levels deep.

Program Counter (PC)

The program counter is eleven bits wide and is organized into 256 byte pages i.e. bits 0 – 7 are part of a counter and bits 8, 9, and 10 are part of a register, loaded under program control, which point to one of eight 256 pages. In addition, some instructions are organized in 64 byte address pages.

Accumulator

The accumulator register is the data path focal point of the CPU. Approximately one-half of the instructions involve the accumulator. Its contents are the source and destination for many ALU operations and port operations. CASE statements (computed GOTOs) are possible by using the Jump with Accumulator (JA) instruction.

Flags

The MSM5840HRS is endowed with the following set of flags.

Z – zero flag	: Indicates that the result of the previous operation was zero
F – all ones	: Indicates a carry from the DP_L register
O – all zeros	: Indicates a borrow from the DP_L register
C – carry	: Indicates a carry from the previous operation
T – timer	: Indicates that the timer/counter is specified as a timer
CT – counter	: Indicates that the timer/counter is specified as a counter
TM – timer flag	: Indicates an overflow of the timer/counter register
INT – interrupt	: Latching memory flag for the external interrupt
INTE – interrupt enable	: Indicates that interrupts have been enabled
H_0 – H_0 memory	: Indicates that an input has been detected on the H_0 input
H_1	: same as H_0 except H_1 input
X	: 0 indicates internal ROM, 1 indicates external ROM. If all external ROM, 0 indicates first bank of 2K.

INSTRUCTION SET

	Mnemonic	Description	Instruction Code 7	6	5	4	3	2	1	0	Byte	Cycle
Load, Store, Read, Clear	CLA	Clear Accumulator	0	0	0	1	0	0	0	0	1	1
	CLL	Clear DP_L	0	0	1	0	0	0	0	0	1	1
	CLH	Clear DP_H	0	1	1	0	0	0	0	0	1	1
	LAI	Load Accumulator with Immediate	0	0	0	1	I_3	I_2	I_1	I_0	1	1
	LLI	Load DP_L with Immediate	0	0	1	0	I_3	I_2	I_1	I_0	1	1
	LHI	Load DP_H with Immediate	0	1	1	0	0	I_2	I_1	I_0	1	1
	L	Load Accumulator with Memory	1	0	0	1	0	1	0	0	1	1
	LM	Load Accumulator with Memory then Modify DP_H	1	0	0	1	0	1	I_1	I_0	1	1
	LAL	Load Accumulator with DP_L	0	1	0	1	0	1	0	1	1	1
	LLA	Load DP_L with Accumulator	0	1	0	1	0	1	0	0	1	1
	LAW	Load Accumulator with W Register	1	0	0	0	0	1	0	0	1	1
	LAX	Load Accumulator with X Register	1	0	0	0	0	1	0	1	1	1
	LAY	Load Accumulator with Y Register	1	0	0	0	0	1	1	0	1	1
	LAZ	Load Accumulator with Z Register	1	0	0	0	0	1	1	1	1	1
	SI	Store Accumulator to Memory then Increment DP_L	1	0	0	1	0	0	0	0	1	1
	SMI	Store Accumulator to Memory then Modify DP_H and Increment DP_L	1	0	0	1	0	0	I_1	I_0	1	1
	LWA	Load W Register with Accumulator	1	0	0	0	0	0	0	0	1	1
	LXA	Load X Register with Accumulator	1	0	0	0	0	0	0	1	1	1
	LYA	Load Y Register with Accumulator	1	0	0	0	0	0	1	0	1	1
	LZA	Load Z Register with Accumulator	1	0	0	0	0	0	1	1	1	1
	LPA	Load Port Pointer with Accumulator	0	1	0	1	1	0	0	0	1	1
	LTI	Load Timer with Immediate	0	1	1	0	1	0	0	0	2	2
			I_7	I_6	I_5	I_4	I_3	I_2	I_1	I_0		
	RTH	Read Timer H	0	1	1	0	1	0	1	0	1	1
	RTL	Read Timer L	0	1	1	0	1	0	1	1	1	1
Exchange	XA	Exchange Accumulator with Save Register A	0	1	0	0	1	0	0	1	1	1
	XL	Exchange DP_L with Save Register L	0	1	0	0	1	0	1	0	1	1
	XCH	Exchange DP_H and Carry with Save Register CH	0	1	0	0	1	0	0	0	1	1
	X	Exchange Accumulator with Memory	1	0	0	1	1	0	0	0	1	1
	XM	Exchange Accumulator with Memory then Modify DP_H	1	0	0	1	1	0	I_1	I_0	1	1
	XAX	Exchange Accumulator with Save Register AX	0	1	0	0	1	0	1	1	1	1
Increment/Decrement	INA	Increment Accumulator	0	0	0	0	0	0	0	1	1	1
	INL	Increment DP_L	0	1	0	1	0	1	1	1	1	1
	INM	Increment Memory	0	1	0	1	1	1	0	1	1	1
	INW	Increment W Register (Skip if Zero)	1	0	0	0	1	0	0	0	1	1
	INX	Increment X Register	1	0	0	0	1	0	0	1	1	1
	INY	Increment Y Register	1	0	0	0	1	0	1	0	1	1
	INZ	Increment Z Register	1	0	0	0	1	0	1	1	1	1

	Mnemonic	Description	Instruction Code								Byte	Cycle
			7	6	5	4	3	2	1	0		
Increment/Decrement	DCA	Decrement Accumulator – Skip if Not All Ones	0	0	0	0	1	1	1	1	1	1
	DCL	Decrement DP_L	0	1	0	1	0	1	1	0	1	1
	DCM	Decrement Memory	0	1	0	1	1	1	0	0	1	1
	DCW	Decrement W Register (Skip if All Ones)	1	0	0	0	1	1	0	0	1	1
	DCX	Decrement X Register	1	0	0	0	1	1	0	1	1	1
	DCY	Decrement Y Register	1	0	0	0	1	1	1	0	1	1
	DCZ	Decrement Z Register	1	0	0	0	1	1	1	1	1	1
	DCH	Decrement DP_H – Skip if All Ones and C = Zero	0	1	0	1	1	1	1	1	1	1
Logical	CAO	Complement Accumulator of One	0	1	0	1	0	0	0	0	1	1
	AND	And Accumulator with Memory	0	1	0	0	0	1	0	0	1	1
	OR	Or Accumulator with Memory	0	1	0	0	0	1	0	1	1	1
	EOR	Exclusive or Accumulator with Memory	0	1	0	0	0	1	1	0	1	1
	RAL	Rotate Accumulator Left through Carry	0	1	0	0	0	1	1	1	1	1
Arithmetic	AC	Add Memory to Accumulator with Carry	0	1	0	0	1	1	0	0	1	1
	ACS	Add Memory to Accumulator with Carry, Skip if Carry	0	1	0	0	1	1	0	1	1	1
	AS	Add Memory to Accumulator, Skip if Carry	0	1	0	0	1	1	1	0	1	1
	AIS	Add Immediate to Accumulator, Skip if Carry	0	0	0	0	I_3	I_2	I_1	I_0	1	1
	DAS	Decimal adjust Accumulator in Subtraction	0	1	0	1	1	0	1	0	1	1
	CM	Compare Accumulator with Memory, Skip if Equal	0	1	0	1	1	1	1	0	1	1
	AWS	Add W Register to Accumulator, Skip if Carry	1	0	0	1	1	1	0	0	1	1
	AYS	Add Y Register to Accumulator, Skip if Carry	1	0	0	1	1	1	1	0	1	1
	AZS	Add Z Register to Accumulator, Skip if Carry	1	0	0	1	1	1	1	1	1	1
Bit Set/Reset/Test	SPB	Set Port Bit	1	0	1	1	0	0	I_1	I_0	1	1
	RPB	Reset Port Bit	1	0	1	1	0	1	I_1	I_0	1	1
	SMB	Set Memory Bit	1	0	1	1	1	0	I_1	I_0	1	1
	RMB	Reset Memory Bit	1	0	1	1	1	1	I_1	I_0	1	1
	TAB	Test Accumulator Bit (Skip if One)	1	0	1	0	0	0	I_1	I_0	1	1
	TMB	Test Memory Bit	1	0	1	0	0	1	I_1	I_0	1	1
	TKB	Test K Port Bit	1	0	1	0	1	0	I_1	I_0	1	1
	THB	Test H Port Bit	1	0	1	0	1	1	1	I_0	1	1
	TI	Test Interrupt flag	1	0	1	0	1	1	1	1	1	1
	TTM	Test Time flag	1	0	1	0	1	1	1	0	1	1
	TC	Test Carry flag	0	1	0	0	0	0	1	0	1	1
	SC	Set Carry flag	0	1	0	0	0	0	0	0	1	1
	RC	Reset Carry flag	0	1	0	0	0	0	0	1	1	1

Mnemonic		Description	Instruction Code								Byte	Cycle
			7	6	5	4	3	2	1	0		
Branch/Subroutine	J	Jump	0	0	1	1	0	I_{10}	I_9	I_8	2	2
			I_7	I_6	I_5	I_4	I_3	I_2	I_1	I_0		
	JC	Jump in Current Page	1	1	I_5	I_4	I_3	I_2	I_1	I_0	1	1
	JA	Jump with Accumulator	0	1	0	0	0	0	1	1	1	1
	CAL	Call Subroutine	0	0	1	1	1	I_{10}	I_9	I_8	2	2
			I_7	I_6	I_5	I_4	I_3	I_2	I_1	I_0		
	RT	Return from Subroutine	0	1	0	1	1	0	0	1	1	2
Input/Output	OBS	Output Byte String	0	1	1	1	0	0	0	0	1	2~17
	OTD	Output Table Data	0	1	1	1	0	0	0	1	1	2
	OA	Output Accumulator to Port A	0	1	1	1	0	0	1	0	1	1
	OB	Output Accumulator to Port B	0	1	1	1	0	0	1	1	1	1
	OP	Output Accumulator to Port P designated Port Pointer	0	1	1	1	0	1	0	0	1	1
	OAB	Output Memory and Accumulator to Port A and B	0	1	1	1	0	1	0	1	1	1
	OPM	Output Memory to Port P designated Port Pointer	0	1	1	1	0	1	1	0	1	1
	IA	Input Port A in Accumulator	0	1	1	1	1	0	1	0	1	1
	IB	Input Port B in Accumulator	0	1	1	1	1	0	1	1	1	1
	IK	Input Port K in Accumulator	0	1	1	1	1	1	0	0	1	1
	IAB	Input Port A and B in Memory and Accumulator	0	1	1	1	1	1	0	1	1	1
Control	EI	Enable Interrupt	0	1	0	1	0	0	1	1	1	1
	DI	Disable Interrupt	0	1	0	1	0	0	1	0	1	1
	ET	Enable Timer	0	1	1	0	1	1	1	1	1	1
	DT	Disable Timer	0	1	1	0	1	1	1	0	1	1
	ECT	Enable Counter	0	1	1	1	1	1	1	1	1	1
	DCT	Disable Counter	0	1	1	1	1	1	1	0	1	1
	HLT	Halt	0	1	1	0	1	1	0	1	1	1
	EXP	Exchange Program	0	1	1	0	1	0	0	1	1	1
	NOP	No Operation	0	0	0	0	0	0	0	0	1	1

ABSOLUTE MAXIMUM RATINGS

Parameter	Symbol	Conditions	Limits	Unit
Supply Voltage	V_{DD}	Ta = 25°C	−0.3 to 7	V
Input Voltage	V_I	Ta = 25°C	−0.3 to V_{DD}	V
Operating Voltage PF PG	V_O	Ta = 25°C	−0.3 to 25	V
Storage Temperature	T_{stg}		−55 to +150	°C

Note: Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or at any other condition above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

OPERATING CONDITIONS

Parameter	Symbol	Conditions	Limits	Unit
Supply Voltage	V_{DD}		3 to 6	V
Operating Temperature	T_{op}		−20 to +70	°C
Fan Out	N	MOS Load	15	
		TTL Load	1	

D.C. CHARACTERISTICS

(V_{DD} = 5V±10%, Ta = −20 to +70°C)

Parameter	Symbol	Conditions	Min.	Typ.	Max.	Unit
High Input Voltage	V_{IH}	—	3.6			V
Low Input Voltage	V_{IL}	—			0.8	V
High Output Voltage (1)	V_{OH}	I_O = −40μA	4.2			V
Low Output Voltage	V_{OL}	I_O = 1.6mA			0.4	V
OSC_0 Input Leak Current	I_{IH}	$V_I = V_{DD}/0V$			25	μA
	I_{IL}				−25	
$\overline{RESET}$, $\overline{MODE}$ Leak Current	I_{IH}	$V_I = V_{DD}/0V$			1	μA
	I_{IL}				−50	
Input Leak Current (2)	I_{IH}	$V_I = V_{DD}/0V$			1	μA
	I_{IL}				−1	
PA, PB High Output Current	I_{OH}	V_{OH} = 0.4V			1	mA
High Output Current (1)	I_{OH}	V_{OH} = 2.5V	−0.25			mA
Low Output Current	I_{OL}	V_{OL} = 0.4V	1.6			mA
PF, PG Output Breakdown Voltage	BV_{OH}	I_O = 10μA	20			V
Input Capacitance	C_I	f = 1MHz Ta = 25°C		5		pF
Output Capacitance	C_O	f = 1MHz Ta = 25°C		7		pF
Current Consumption (3)	I_{DD}	$V_I = V_{DD}/0V$		20	200	μA
	I_{DD}	$V_I = V_{DD}/0V$ f = 4.2MHz		1.6	4	mA

Notes: (1) Except PA, PB (see graphs)
(2) Except OSC_0, $\overline{RESET}$, $\overline{MODE}$
(3) Typical Value of V_{DD} is 5V

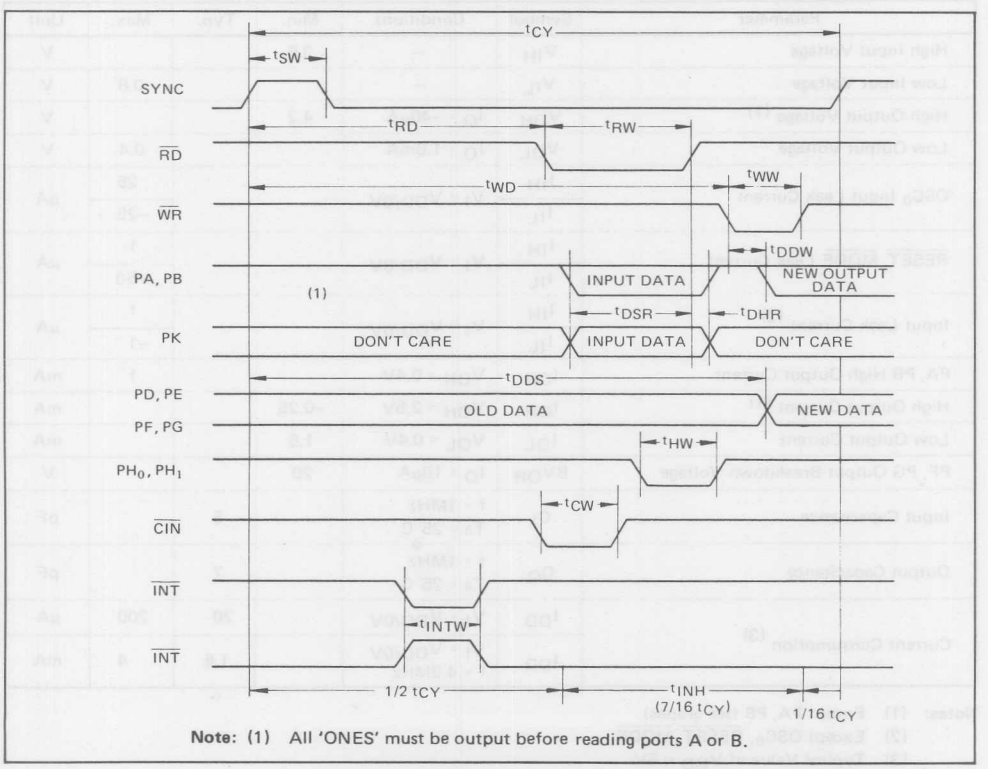
A.C. CHARACTERISTICS (INTERNAL ROM MODE)

(V_{DD} = 5V±10%, Ta = −20° to +70°C)

Parameter	Symbol	Conditions	Min.	Typ.	Max.	Unit
Cycle Time	t_{CY}		7.6			μS
Sync Pulse Width	t_{SW}		0.95			μS
$\overline{RD}$ Pulse Width	t_{RW}		1.9			μS
Sync ↑ to $\overline{RD}$ ↓	t_{RD}	C_L = 50pF	1/2 t_{CY} + 0.5			μS
$\overline{WR}$ Pulse Width	t_{WW}		0.95			μS
Sync ↑ to $\overline{WR}$ ↓	t_{WD}	C_L = 50pF	13/16 t_{CY}+0.5			μS
Port Input Setup Time	t_{DSR}		4/16 t_{CY}			μS
Port Input Hold Time	t_{DHR}		0		0.8	μS
$\overline{WR}$↓ to New Data Valid	t_{DDW}	PA, PB C_L = 50pF			0.8	μS
Sync↑ to New Data Valid	t_{DDS}	PD, PE, PF, PG C_L = 50pF			13/16 t_{CY}+0.5	μS
PH_0, PH_1 Input Pulse Width	t_{HW}	(1)	500			nS
$\overline{CIN}$ Input Pulse Width	t_{CW}		250			nS
INT Input Pulse Width	t_{INTW}	(1)	500			nS

Note: (1) The processor logic will ignore the following events:

1. An INT falling edge occurring during T_{INH} of a T_I instruction.
2. A PH_0 or PH_1 low level occuring only during T_{INH} of a T_{HB} instruction.

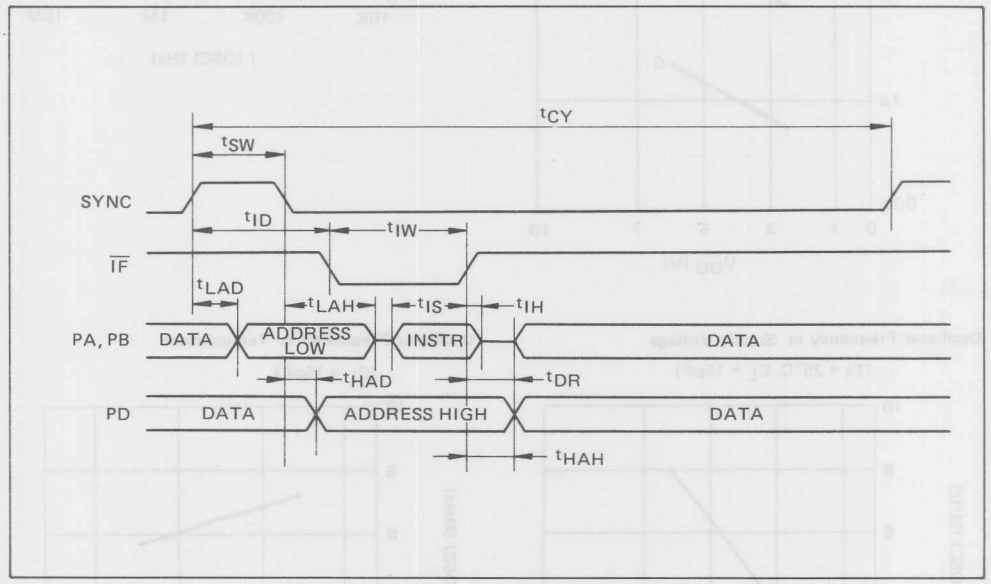


Note: (1) All 'ONES' must be output before reading ports A or B.

A.C. CHARACTERISTICS (EXTERNAL ROM MODE)

(V_{DD} = 5V±10%, Ta = −20° to +70°C)

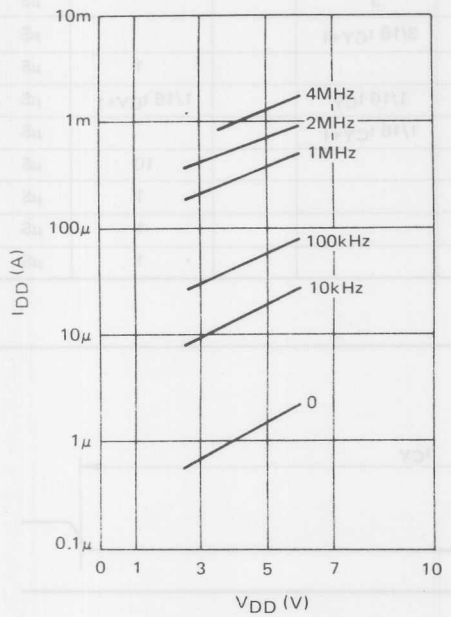
Parameter	Symbol	Conditions	Min.	Typ.	Max.	Unit
Cycle Time	t_{CY}		7.6			μS
Sync Pulse Width	t_{SW}		0.95			μS
$\overline{IF}$ Pulse Width	t_{IW}		3			μS
Sync↑ to $\overline{IF}$↓	t_{ID}	C_L = 50pF	3/16 t_{CY}+1			μS
Address Low Delay	t_{LAD}	C_L = 50pF			1	μS
Address Low Hold	t_{LAH}		1/16 t_{CY}		1/16 t_{CY}+1	μS
Instruction Setup	t_{IS}		1/16 t_{CY}+1			μS
Instruction Hold	t_{IH}				10	μS
Data Recovery	t_{DR}	C_L = 50pF			1	μS
Address High Delay	t_{HAD}	C_L = 50pF			1	μS
Address High Hold	t_{HAH}				1	μS



Cycle Dependent Timings	@4MHz	2MHz	1MHz	500kHz
1/16 t_{CY}	0.5μS	1μS	2μS	4μS
1/16 t_{CY} + 1	1.5μS	2μS	3μS	5μS
3/16 t_{CY} + 1	2.5μS	4μS	7μS	13μS
4/16 t_{CY} − 1	1μS	3μS	7μS	15μS
1/2 t_{CY} + 1	5μS	9μS	17μS	33μS
7/16 t_{CY}	3.5μS	7μS	14μS	28μS
13/16 t_{CY} + 1	7.5μS	14μS	27μS	53μS

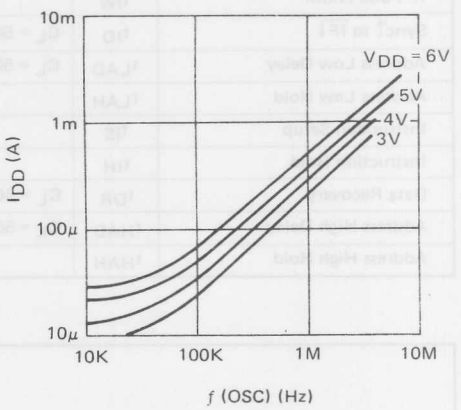
Supply Current vs Supply Voltage

($Ta = 25^{\circ}C$, $C_L = 15pF$)



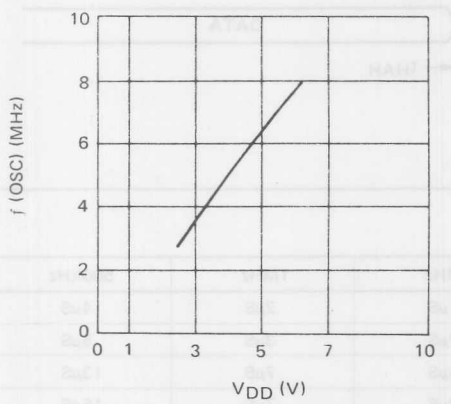
Supply Current vs Oscillator Frequency

($Ta = 25^{\circ}C$, $C_L = 15pF$)



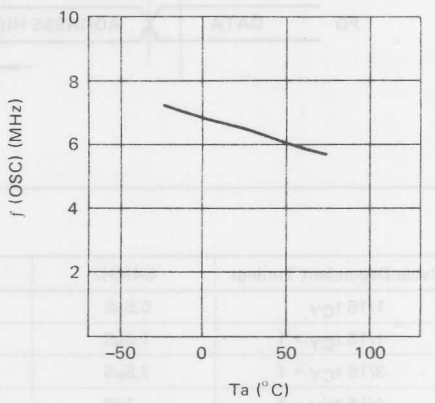
Oscillator Frequency vs Supply Voltage

($Ta = 25^{\circ}C$, $C_L = 15pF$)

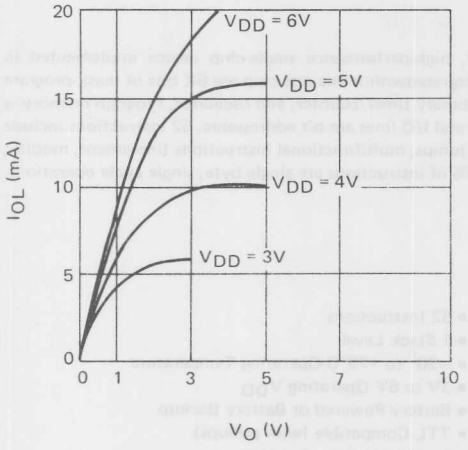


Oscillator Frequency vs Temperature

($C_L = 15pF$)

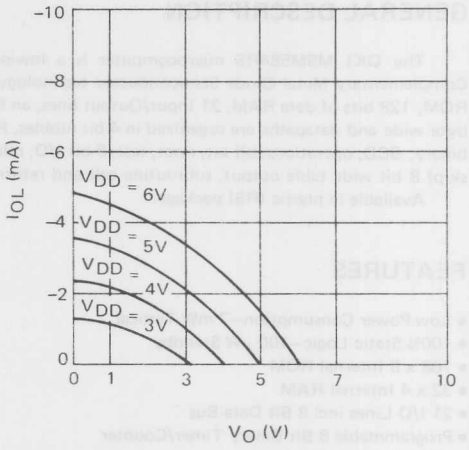


Low Current Out vs Voltage



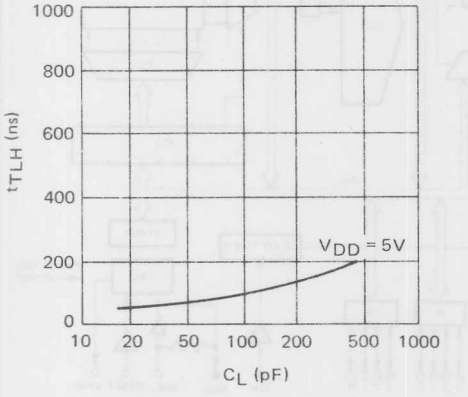
High Current Out vs Voltage

(Ta = 25°C, Except PA, PB)



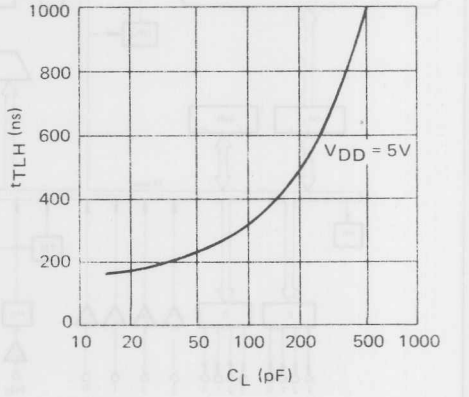
Fall Time vs Load

(Ta = 25°C, PA, PB, PD, PE, $\overline{RD}$, $\overline{WR}$, $\overline{IF}$, SYNC)



Rise Time vs Load

(Ta = 25°C, PD, PE, $\overline{RD}$, $\overline{WR}$, $\overline{IF}$, SYNC)



OKI semiconductor

MSM5842RS

CMOS SINGLE COMPONENT MICROCOMPUTER

GENERAL DESCRIPTION

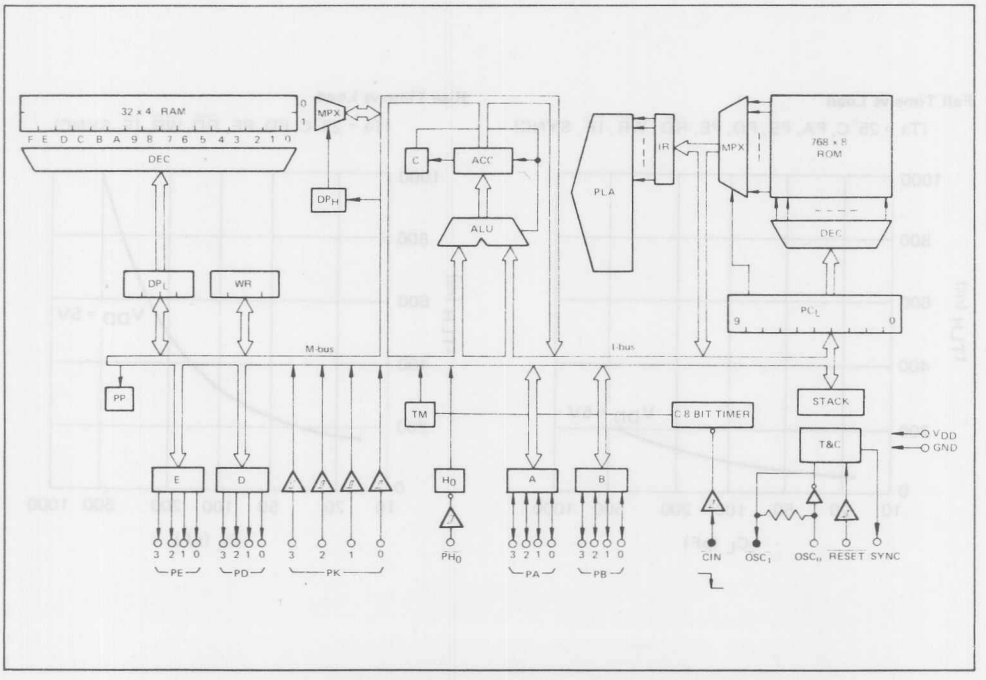
The OKI MSM584RS microcomputer is a low-power, high-performance single-chip device implemented in Complementary Metal Oxide Semiconductor technology. Integratedwithin this one chip are 6K bits of mask program ROM, 128 bits of data RAM, 21 Input/Output lines, an 8-bit binary timer/counter, and oscillator. Program memory is byte wide and datapaths are organized in 4 bit nibbles. RAM and I/O lines are bit addressable. 52 instructions include binary, BCD, operations; bit set, reset, test; 8 bit I/O; relative jumps; multifunctional instructions (increment, modify, skip) 8 bit wide table output; subroutine call and return. 94% of instructions are single byte, single cycle operations.

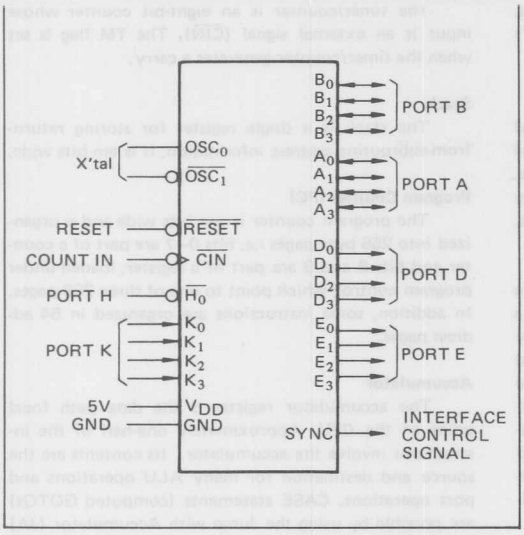
Available in plastic (RS) package.

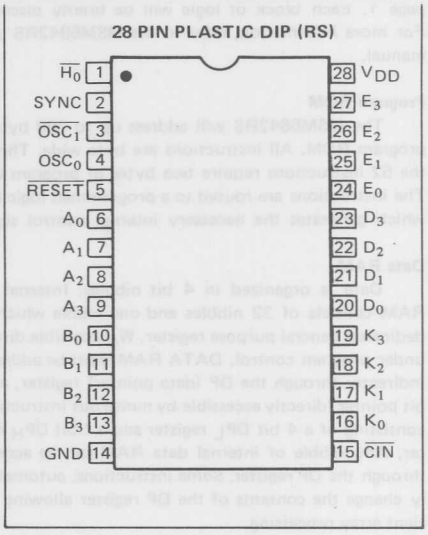
FEATURES

- Low Power Consumption—7 mW Typical
- 100% Static Logic—100 μW Standby
- 768 x 8 Internal ROM
- 32 x 4 Internal RAM
- 21 I/O Lines incl 8 Bit Data Bus
- Programmable 8 Bit Binary Timer/Counter
- Self-contained Oscillator
- 52 Instructions
- 1 Stack Level
- −20° to +70° C Operating Temperature
- 3V to 6V Operating V_{DD}
- Battery Powered or Battery Backup
- TTL Compatible (with pullups)
- 8 μS Cycle Time @ 4 MHz

FUNCTIONAL BLOCK DIAGRAM







PIN DESCRIPTION

Designation	Pin No.	Function
GND	14	Circuit GND potential
V_{DD}	28	Main power source (+5V)
OSC_0	4	Crystal OSC input, external clock input
$\overline{OSC_1}$	3	Crystal OSC input, external clock output (not TTL compatible)
PA,PB	6 to 13	Quasi-bidirectional ports for 4-bit parallel I/O. Used as a pair for 8 bit I/O.
PD, PE	20 to 27	Output ports for 4 bit parallel output and bit set/reset. Specified by internal port pointer. Bit position specified by set/reset instruction.
PK	16 to 19	4 bit parallel or bit test input port (unlatched)
$\overline{PH}$	1	1 bit input port with latched memory (negative level sensitive)
$\overline{RESET}$	5	RESET must be low for more than one machine cycle and has priority over every other signal. (see MSM5842 user's manual for initialization sequence)
$\overline{CIN}$	15	Negative edge sensitive external input for timer/counter.
SYNC	2	General purpose synchronizing signal output at the beginning of each machine cycle.

FUNCTIONAL DESCRIPTION

A block diagram of the MSM5842RS is given on page 1. Each block of logic will be briefly discussed. For more information, refer to the MSM5842RS user's manual.

Program ROM

The MSM5842RS will address up to 768 bytes of program ROM. All instructions are byte wide. Three of the 52 instructions require two bytes of program code. The instructions are routed to a programmed logic array which generates the necessary internal control signals.

Data RAM

Data is organized in 4 bit nibbles. Internal data RAM consists of 32 nibbles and one nibble which is a dedicated general purpose register, W, accessible directly under program control. DATA RAM must be addressed indirectly through the DP (data pointer) register, a five bit pointer (directly accessible by numerous instructions) consisting of a 4 bit DP_L register and a 1 bit DP_H register. Any nibble of internal data RAM can be accessed through the DP register. Some instructions, automatically change the contents of the DP register allowing efficient array processing.

Input/Output Ports

PA, PB—These two ports are quasi-bidirectional ports which can be used as simple I/O lines or used as either 4 bit or 8 bit parallel bus.

PD, PE—These two output ports are addressed indirectly through the ONE BIT port pointer whose contents are changed through certain instructions. These ports are bit (set/reset) addressable.

PK is an input port without memory, addressable as a nibble input.

PH is a one bit input port with memory, which can be tested and reset under program control.

Timer/Counter

The timer/counter is an eight-bit counter whose input is an external signal ($\overline{CIN}$). The TM flag is set when the timer/counter generates a carry.

Stack

The stack is a single register for storing return-from-subroutine address information. It is ten bits wide.

Program Counter (PC)

The program counter is ten bits wide and is organized into 256 byte pages i.e. bits 0—7 are part of a counter and bits 8 and 9 are part of a register, loaded under program control, which point to one of three 256 pages. In addition, some instructions are organized in 64 address pages.

Accumulator

The accumulator register is the data path focal point of the CPU. Approximately one-half of the instructions involve the accumulator. Its contents are the source and destination for many ALU operations and port operations. CASE statements (computed GOTOs) are possible by using the Jump with Accumulator (JA) instruction.

Flags

The MSM5842RS is endowed with the following set of flags.

Z —zero flag: Indicates that the result of the previous operation was zero

C —carry: Indicates a carry from the previous operation

TM —timer flag: Indicates an overflow of the timer/counter register

H_0 —H_0 memory: Indicates that an input has been detected on the H_0 input

INSTRUCTION SET

Mnemonic		Description	Instruction Code								Byte	Cycle
			7	6	5	4	3	2	1	0		
Load, Store, Read, Clear	CLA	Clear Accumulator	0	0	0	1	0	0	0	0	1	1
	CLL	Clear DP_L	0	0	1	0	0	0	0	0	1	1
	CLH	Clear DP_H	0	1	1	0	0	0	0	0	1	1
	LAI	Load Accumulator with Immediate	0	0	0	1	I_3	I_2	I_1	I_0	1	1
	LLI	Load DP_L with Immediate	0	0	1	0	I_3	I_2	I_1	I_0	1	1
	LHI	Load DP_H with Immediate	0	1	1	0	0	0	0	I_0	1	1
	L	Load Accumulator with Memory	1	0	0	1	0	1	0	0	1	1
	LAL	Load Accumulator with DP_L	0	1	0	1	0	1	0	1	1	1
	LLA	Load DP_L with Accumulator	0	1	0	1	0	1	0	0	1	1
	LAW	Load Accumulator with W Register	1	0	0	0	0	1	0	0	1	1
	SI	Store Accumulator to Memory then Increment DP_L	1	0	0	1	0	0	0	0	1	1
	LWA	Load W Register with Accumulator	1	0	0	0	0	0	0	0	1	1
	LPA	Load Port Pointer with Accumulator	0	1	0	1	1	0	0	0	1	1
	LTI	Load Timer with Immediate—All Zeros	0	1	1	0	1	0	0	0	1	1

	Mnemonic	Description	Instruction Code 7	6	5	4	3	2	1	0	Byte	Cycle
Exchange	X	Exchange Accumulator with Memory	1	0	0	1	1	0	0	0	1	1
Increment/ Decrement	INA	Increment Accumulator } Skip if Zero	0	0	0	0	0	0	0	1	1	1
	INL	Increment DP_L	0	1	0	1	0	1	1	1	1	1
	INM	Increment Memory	0	1	0	1	1	1	0	1	1	1
	INW	Increment W Register	1	0	0	0	1	0	0	0	1	1
	DCA	Decrement Accumulator—Skip if Not All Ones	0	0	0	0	1	1	1	1	1	1
	DCL	Decrement DP_L } Skip if All Ones	0	1	0	1	0	1	1	0	1	1
	DCM	Decrement Memory	0	1	0	1	1	1	0	0	1	1
Logical	CAO	Complement Accumulator of One	0	1	0	1	0	0	0	0	1	1
	RAL	Rotate Accumulator Left through Carry	0	1	0	0	0	1	1	1	1	1
Arithmetic	AC	Add Memory to Accumulator with Carry	0	1	0	0	1	1	0	0	1	1
	AS	Add Memory to Accumulator, Skip if Carry	0	1	0	0	1	1	1	0	1	1
	AIS	Add Immediate to Accumulator, Skip if Carry	0	0	0	0	I_3	I_2	I_1	I_0	1	1
	DAS	Decimal adjust Accumulator In Subtraction	0	1	0	1	1	0	1	0	1	1
	CM	Compare Accumulator with Memory, Skip if Equal	0	1	0	1	1	1	1	0	1	1
Bit Set/Reset/Test	SMB	Set Memory Bit	1	0	1	1	1	0	I_1	I_0	1	1
	RMB	Reset Memory Bit	1	0	1	1	1	1	I_1	I_0	1	1
	TAB	Test Accumulator Bit	1	0	1	0	0	0	I_1	I_0	1	1
	TMB	Test Memory Bit	1	0	1	0	0	1	I_1	I_0	1	1
	THB	Test H Port Bit	1	0	1	0	1	1	1	I_0	1	1
	TTM	Test Time flag	1	0	1	0	1	1	1	0	1	1
	TC	Test Carry flag	0	1	0	0	0	0	1	0	1	1
	SC	Set Carry flag	0	1	0	0	0	0	0	0	1	1
	RC	Reset Carry flag	0	1	0	0	0	0	0	1	1	1
Branch/Subroutine	J	Jump	0	0	1	1	0	0	I_9	I_8	2	2
			I_7	I_6	I_5	I_4	I_3	I_2	I_1	I_0		
	JC	Jump in Current Page	1	1	I_5	I_4	I_3	I_2	I_1	I_0	1	1
	JA	Jump with Accumulator	0	1	0	0	0	0	1	1	1	1
	CAL	Call Subroutine	0	0	1	1	1	0	I_9	I_8	2	2
			I_7	I_6	I_5	I_4	I_3	I_2	I_1	I_0		
	RT	Return from Subroutine	0	1	0	1	1	0	0	1	1	2
Input/Output	OTD	Output Table Data	0	1	1	1	0	0	0	1	1	2
	OA	Output Accumulator to Port A	0	1	1	1	0	0	1	0	1	1
	OB	Output Accumulator to Port B	0	1	1	1	0	0	1	1	1	1
	OP	Output Accumulator to Port P designated Port Pointer	0	1	1	1	0	1	0	0	1	1
	OPM	Output Memory to Port P designated Port Pointer	0	1	1	1	0	1	1	0	1	1
	IA	Input Port A in Accumulator	0	1	1	1	1	0	1	0	1	1
	IB	Input Port B in Accumulator	0	1	1	1	1	0	1	1	1	1
	IK	Input Port K in Accumulator	0	1	1	1	1	1	0	0	1	1
Control	NOP	No Operation	0	0	0	0	0	0	0	0	1	1

ABSOLUTE MAXIMUM RATINGS

Parameter	Symbol	Conditions	Limits	Unit
Supply Voltage	V_{DD}	Ta = 25°C	−0.3 to 7	V
Input Voltage	V_I	Ta = 25°C	−0.3 to V_{DD}	V
Storage Temperature	T_{stg}		−55 to +150	°C

Note: Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or at any other condition above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

OPERATING CONDITIONS

Parameter	Symbol	Conditions	Limits	Unit
Supply Voltage	V_{DD}		3 to 6	V
Operating Temperature	T_{op}		−20 to +70	°C
Fan Out		MOS Load	40	
		TTL Load	1	

D.C. CHARACTERISTICS

(V_{DD} = 5V±10%, Ta = −20 to +70°C)

Parameter	Symbol	Conditions	Min.	Typ.	Max.	Unit
High Input Voltage	V_{IH}	–	3.6			V
Low Input Voltage	V_{IL}	–			0.8	V
High Output Voltage(1)	V_{OH}	I_O = −40 μA	4.2			V
Low Output Voltage	V_{OL}	I_O = 1.6 mA			0.45	V
OSC_0 Input Leak Current	I_{IH} I_{IL}	$V_I = V_{DD}/0V$			25 −25	μA
$\overline{RESET}$ Leak Current	I_{IH} I_{IL}	$V_I = V_{DD}/0V$			1 −20	μA
Input Leak Current(2)	I_{IH} I_{IL}	$V_I = V_{DD}/0V$			1 −1	μA
PA, PB High Output Current	I_{OH}	V_{OH} = 0.4V			1	mA
High Output Current(1)	I_{OH}	V_{OH} = 2.5V	−0.25			mA
Low Output Current	I_{OL}	V_{OL} = 0.45V	1.6			mA
PF, PG Output Breakdown Voltage	BV_{OH}	I_O = 10 μA	20			V
Input Capacitance	C_I	f = 1 MHz Ta = 25°C		5		pF
Output Capacitance	C_O	f = 1 MHz Ta = 25°C		7		pF
Current Consumption(3)	I_{DD}	$V_I = V_{DD}/0V$		20	200	μA
	I_{DD}	$V_I = V_{DD}/0V$ f = 4 MHz		1.5	4	mA

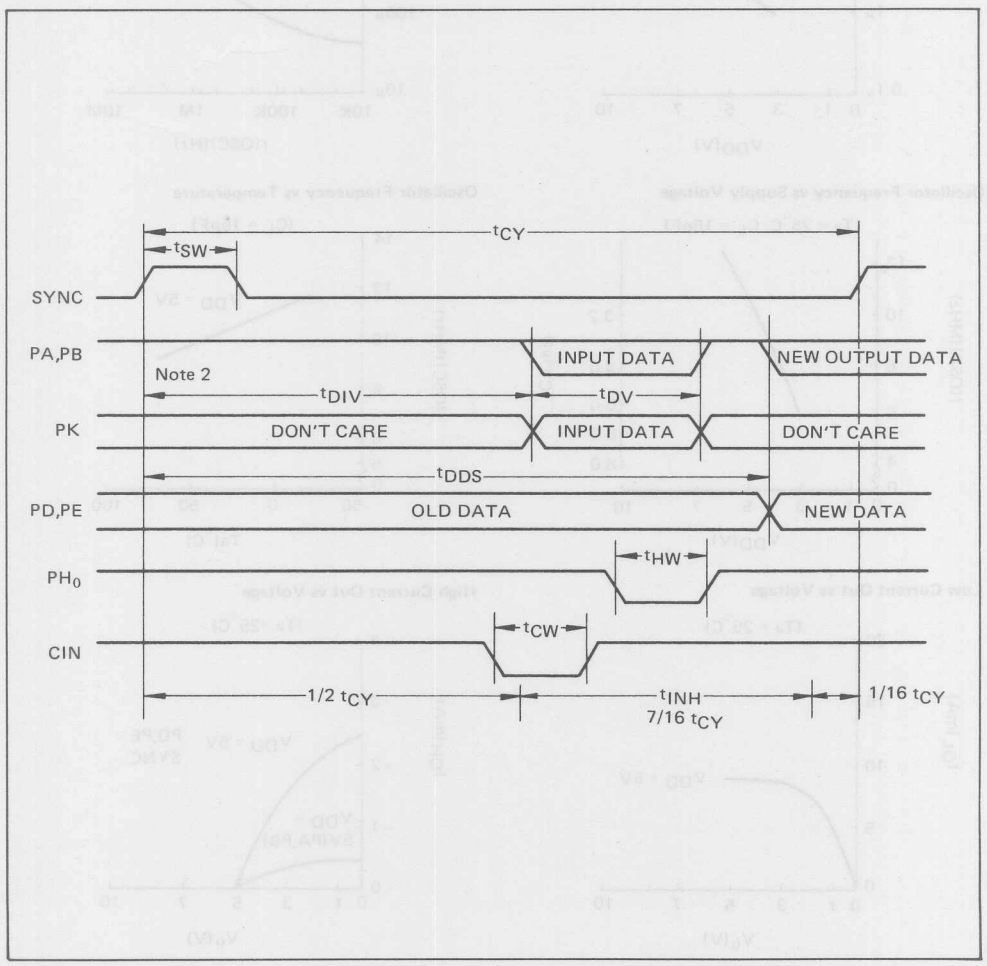
Notes: (1) Except PA, PB (see graphs)
(2) Except OSC_0, $\overline{RESET}$
(3) Typical Value of V_{DD} is 5V

A.C. CHARACTERISTICS

(V_{DD} = 5V±10%, Ta = –20° to + 70°C, $f_{(OSC)}$ = 4 MHz)

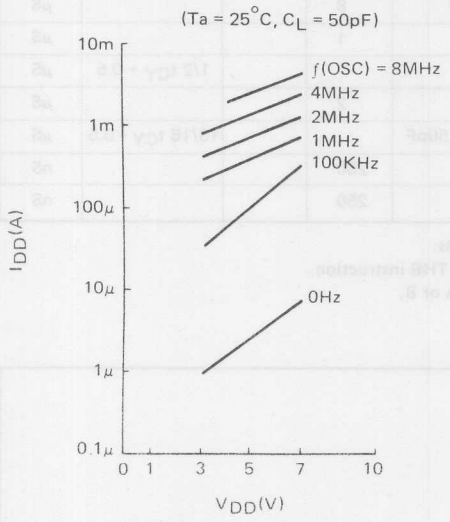
Parameter	Symbol	Conditions	Min.	Typ.	Max.	Unit
Cycle Time	t_{CY}	O_{SC} = 4 MHz	8			µS
Sync Pulse Width	t_{SW}		1			µS
Port Input Invalid Time	t_{DIV}				1/2 t_{CY} + 0.5	µS
Port Input Valid Time	t_{DV}		2			µS
Sync↑ to New Data Valid	t_{DDS}	PD, PE C_L = 50pF			13/16 t_{CY} + 0.5	µS
PH_0 Input Pulse Width	t_{HW}	(1)	250			nS
$\overline{CIN}$ Input Pulse Width	t_{CW}		250			nS

Notes: (1) The processor logic will ignore the following events:
A PH_0 low level occurring only during T_{INH} of a THB instruction.
(2) All 'ONES' must be output before reading ports A or B.

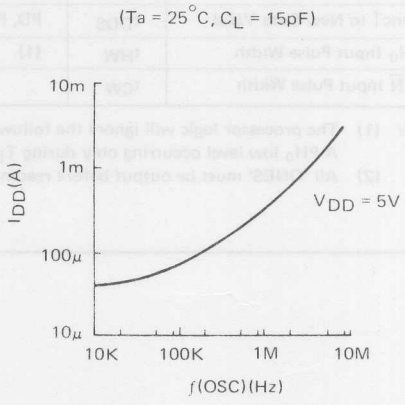


TYPICAL PERFORMANCE CURVES

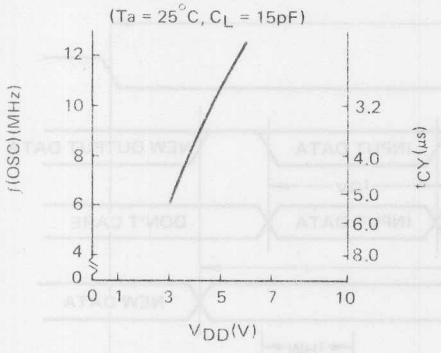
Supply Current vs Supply Voltage



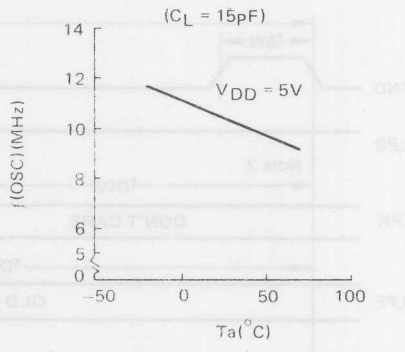
Supply Current vs Oscillator Frequency



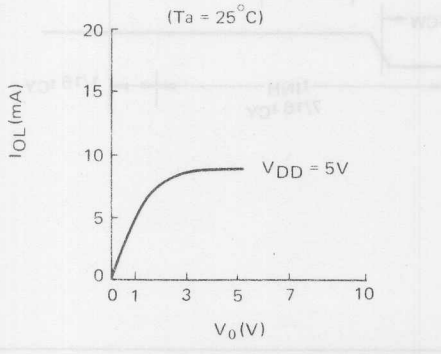
Oscillator Frequency vs Supply Voltage



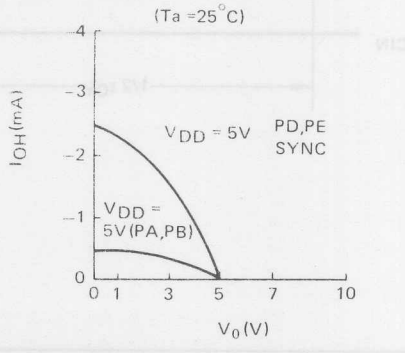
Oscillator Frequency vs Temperature

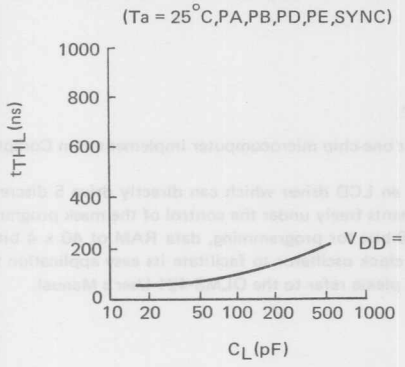


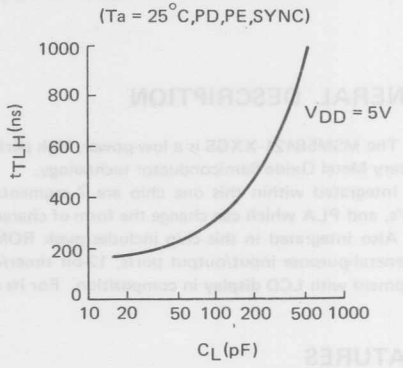
Low Current Out vs Voltage



High Current Out vs Voltage







OLMS–40 FAMILY TOOLS

The MSM5842RS single-chip microcomputer is but one device in a family of microcomputers and development tools. The other microcomputers are basically sub-sets and super sets of the MSM5840RS with more or less memory, I/O, and package pins and are covered in their respective data sheets. In addition to the other family members, there is an extensive set of development and debug tools available to the designer. These include:

SDP–40x: Software development package consisting of a disk based assembler and down-loading utilities compatible with either ISIS® or CP/M® operating systems (SDP–401 for ISIS, SPD–40C for CP/M) available on single or Intel standard double density floppy diskettes. ISIS is a registered trademark of Intel Corp. CP/M is a registered trademark of Digital Research Corp.

MPB–201: This PC board can be used as a stand-alone development tool complete with ROM based self-assembler and debugging monitor. Programs can be executed from PROM or on-board RAM loaded from an ASR–33 (or equivalent) which is used for I/O of programs in the stand alone mode.

This PC board also contains a full bus interfacing capability so that programs may be assembled on a development system and either down-loaded to the on-board RAM or executed directly from the development system memory.

Battery backup exists on board so that the on-board RAM contents can be saved for a period of many days without the application of power. Thus, frequent saving of the developing program is unnecessary.

MPB–202: This PC board contains an MSM5840RS microcomputer and two sockets for 2716 EPROMS with the necessary additional logic to emulate the operation of an internal ROM based MSM5840RS. Unlike MPB–201 board, it does not contain a keyboard, display, on-board, RAM, or capability of interfacing to a development system. It is intended as a final stand-alone verification step before submitting the program for masking.

MPB–203: This PC board is designed to interface to an MPB–201 board and contains the logic necessary for interfacing the MPB–201 CMOS logic signals to an EIA interface, a 20 ma loop interface, or a TTL compatible interface for serial asynchronous ASCII communications with an ASR–33 (or equivalent) when the MPB–201 is used in stand-alone mode i.e. not connected to a development system.

This board also contains a socket for EPROMs so that 2716s can be programmed directly from the MPB–201 board without the need of an external PROM programmer. The associated software is contained in the MPB–201 monitor.

The MSM5842RS can be emulated by connecting the MPB–201 to the MPB–203 and using the output connector on the MPB–203 to connect into the users MSM5842RS socket.

A truly stand-alone development tool results from the combination of the MPB–201 and MPB–203.

OKI semiconductor

MSM58421GS

4-BIT ONE CHIP MICROCOMPUTER WITH LCD DRIVER

GENERAL DESCRIPTION

The MSM58421-XXGS is a low-power, high performance 4-bit one-chip microcomputer implemented in Complementary Metal Oxide Semiconductor technology.

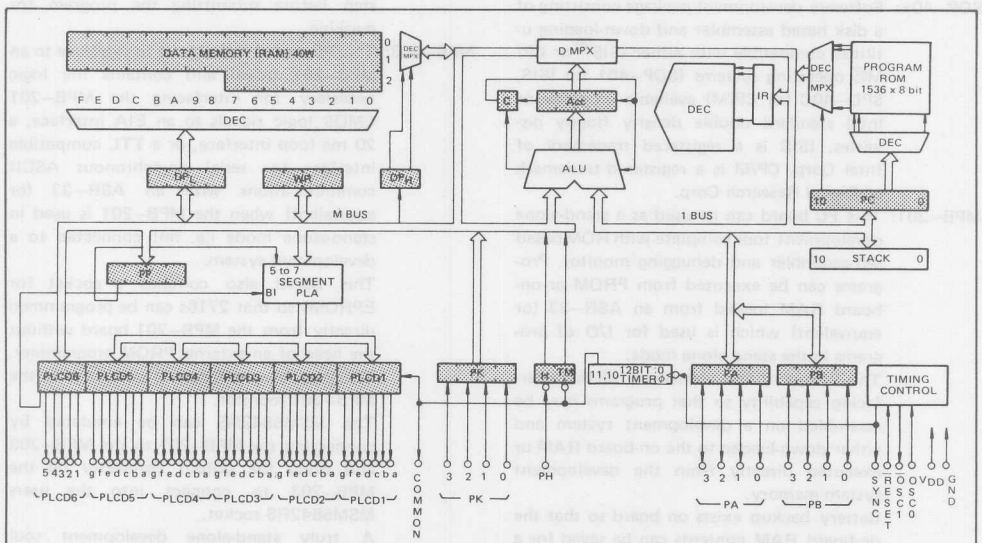
Integrated within this one chip are 7 segments in 5 digits, an LCD driver which can directly drive 5 discrete LCD's, and PLA which can change the form of character of 7 segments freely under the control of the mask program.

Also integrated in this chip includes mask ROM of 1536 x 8 bits for programming, data RAM of 40 x 4 bits, 13 general-purpose input/output ports, 12-bit timer/counter, and clock oscillator to facilitate its easy application to equipment with LCD display in composition. For its employment, please refer to the OLMS-421 User's Manual.

FEATURES

- Low Power Consumption CMOS 4-Bit One-Chip Microcomputer
- 100% Static Logic
- 1536 x 8 Bits Mask ROM
- 40 x 4 Bits Data RAM
- 1 Static Register
- Built-in 12-Bit Timer/Counter (with 32 Hz Common Output)
- Equipped with All Input Ports Schmitt Trigger Circuit
- 8-Bit Interface Function
- 52 Instructions
- 94% of 52 Instructions is of a 1 Byte and 1 Machine Cycle Type
- Integrated with 13 Input/Output Ports and 40 Static LCD Driver Circuit
- +5V Single Power Supply, 60-Pin Mold Flat Package
- 7-Segment Character Form Programmable Freely by User (32 Words x 7 Segments)
- Various Functions Changeable under Mask Program Control

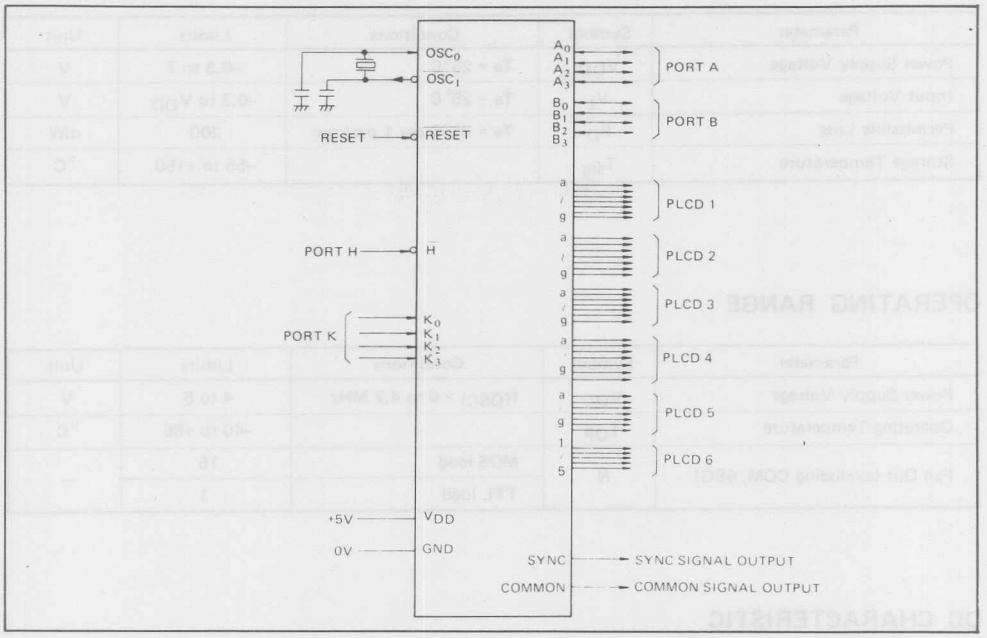
FUNCTIONAL BLOCK DIAGRAM



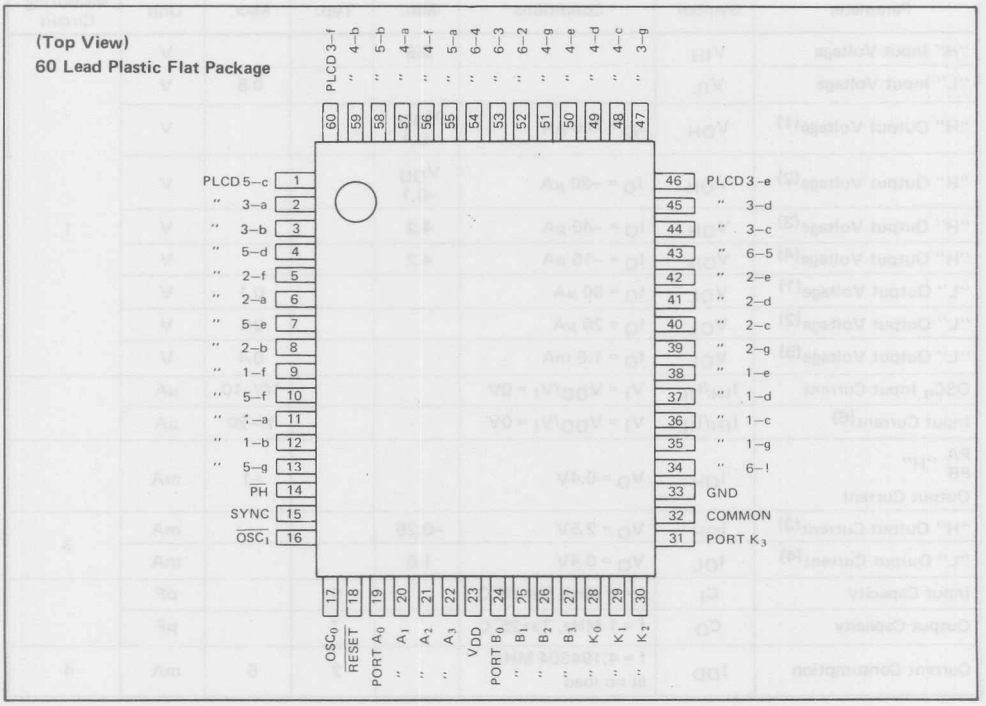
* Crystal Oscillator 4.194304 MHz, SYNC ≒ 7.63 μsec Common 32 Hz

* Portions operable by ▨ instruction.

LOGIC SYMBOL



PIN CONFIGURATION



ABSOLUTE MAXIMUM RATINGS

Parameter	Symbol	Conditions	Limits	Unit
Power Supply Voltage	V_{DD}	Ta = 25°C	−0.3 to 7	V
Input Voltage	V_I	Ta = 25°C	−0.3 to V_{DD}	V
Permissible Loss	P_D	Ta = 25°C per 1 package	200	mW
Storage Temperature	T_{stg}		−55 to +150	°C

OPERATING RANGE

Parameter	Symbol	Conditions	Limits	Unit
Power Supply Voltage	V_{DD}	$f_{(OSC)}$ = 0 to 4.2 MHz	4 to 6	V
Operating Temperature	T_{OP}	–	−40 to +85	°C
Fan Out (excluding COM, SEG)	N	MOS load	15	–
		TTL load	1	

DC CHARACTERISTIC

(V_{DD} = 5V±10%, Ta = −40 to +85°C)

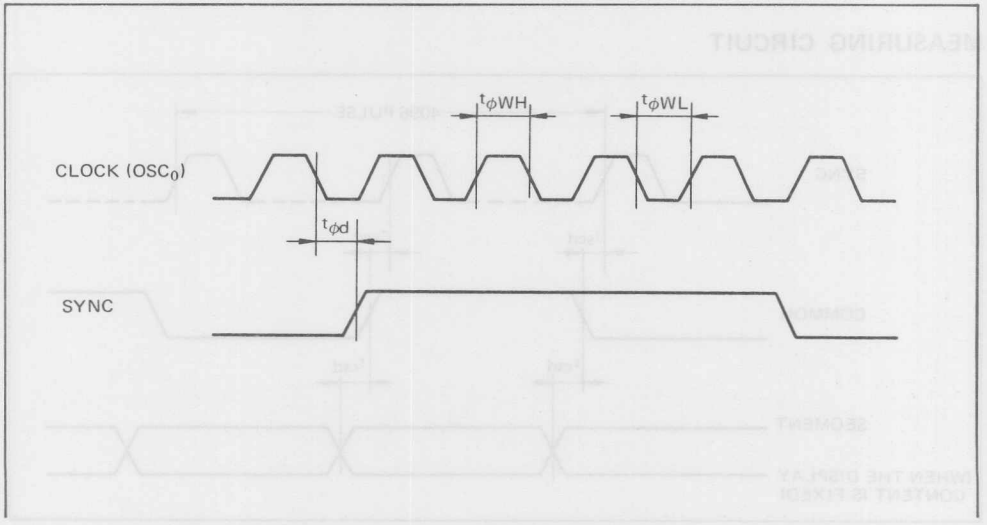
Parameter	Symbol	Conditions	Min.	Typ.	Max.	Unit	Measuring Circuit
"H" Input Voltage	V_{IH}		3.6			V	1
"L" Input Voltage	V_{IL}				0.8	V	
"H" Output Voltage(1)	V_{OH}	I_O = −80 μA	V_{DD} −0.1			V	
"H" Output Voltage(2)	V_{OH}	I_O = −20 μA	V_{DD} −0.1			V	
"H" Output Voltage(3)	V_{OH}	I_O = −40 μA	4.2			V	
"H" Output Voltage(4)	V_{OH}	I_O = −15 μA	4.2			V	
"L" Output Voltage(1)	V_{OL}	I_O = 80 μA			0.1	V	
"L" Output Voltage(2)	V_{OL}	I_O = 20 μA			0.1	V	
"L" Output Voltage(5)	V_{OL}	I_O = 1.6 mA			0.4	V	
OSC_0 Input Current	I_{IH}/I_{IL}	$V_I = V_{DD}/V_I$ = 0V			10/−10	μA	
Input Current(6)	I_{IH}/I_{IL}	$V_I = V_{DD}/V_I$ = 0V			1/−20	μA	
PA, PB "H" Output Current	I_{OH}	V_O = 0.4V			−1	mA	
"H" Output Current(3)	I_{OH}	V_O = 2.5V	−0.25			mA	3
"L" Output Current(4)	I_{OL}	V_O = 0.4V	1.6			mA	
Input Capacity	C_I	f = 1 MHz, Ta=25°C		5		pF	
Output Capacity	C_O	f = 1 MHz, Ta=25°C		7		pF	
Current Consumption	I_{DD}	f = 4.194304 MHz, at no load		2	5	mA	4

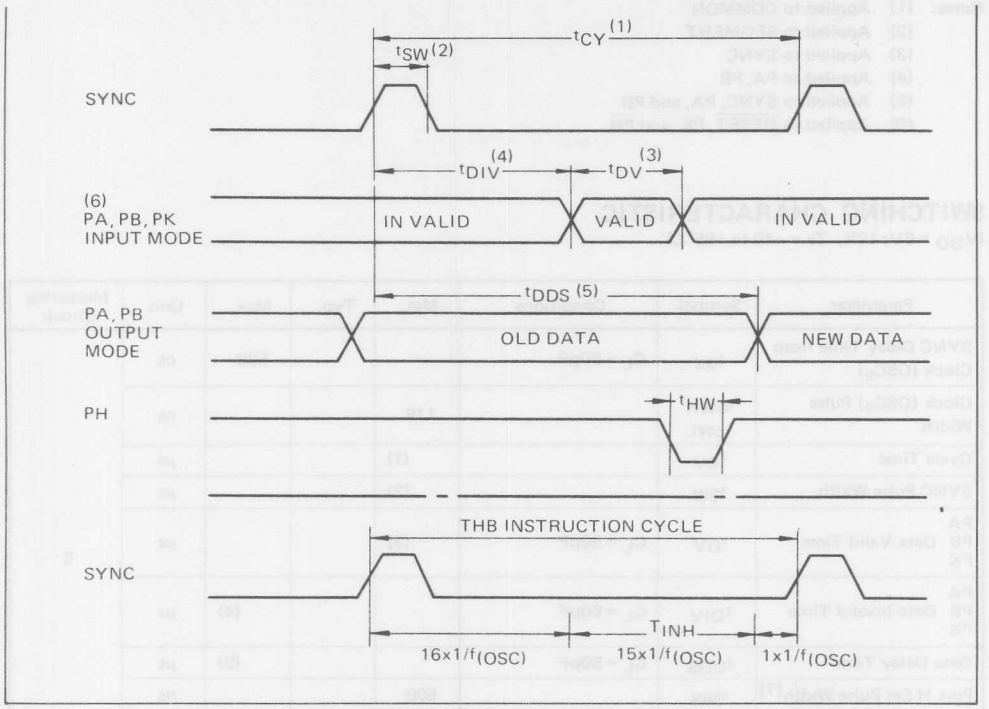
(5) Applied to $\overline{\text{SYNC}}$, PA, and $\overline{\text{PB}}$
(6) Applied to $\overline{\text{RESET}}$, PK, and $\overline{\text{PH}}$

SWITCHING CHARACTERISTIC

(V_{DD} = 5V±10%, Ta = −40 to +85°C)

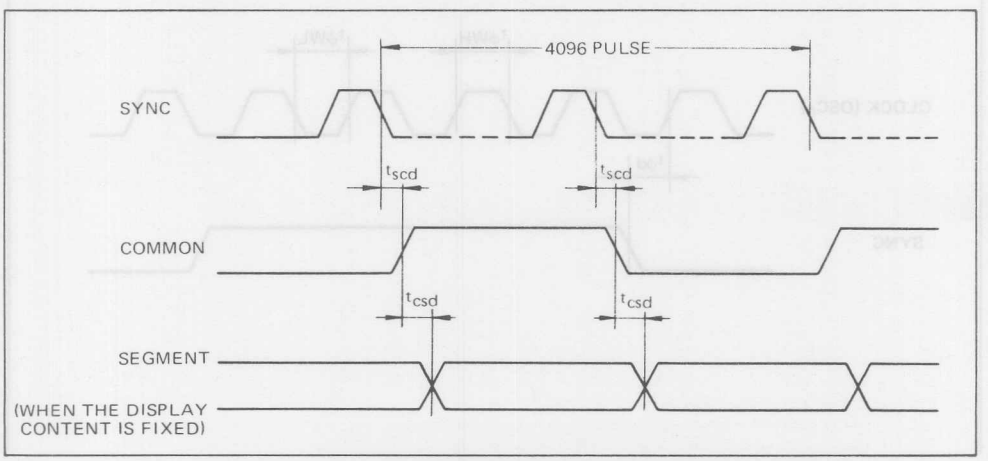
Parameter	Symbol	Conditions	Min.	Typ.	Max.	Unit	Measuring Circuit
SYNC Delay Time from Clock (OSC_0)	$t_{\phi d}$	C_L = 50pF			800	ns	5
Clock (OSC_0) Pulse Width	$t_{\phi WH}$ $t_{\phi WL}$		115			ns	
Cycle Time	t_{CY}		(1)			μs	
SYNC Pulse Width	t_{SW}		(2)			μs	
PA PB Data Valid Time PK	t_{DV}	C_L = 50pF	(3)			μs	
PA PB Data Invalid Time PK	t_{DIV}	C_L = 50pF			(4)	μs	
Data Delay Time	t_{DDS}	C_L = 50pF			(5)	μs	
Port H Set Pulse Width(7)	t_{HW}		500			ns	
COMMON Delay Time from SYNC	t_{SCd}	C_L = 50pF			2	μs	
SEGMENT Delay Time from COMMON	t_{CSd}	C_L = 50pF			1	μs	

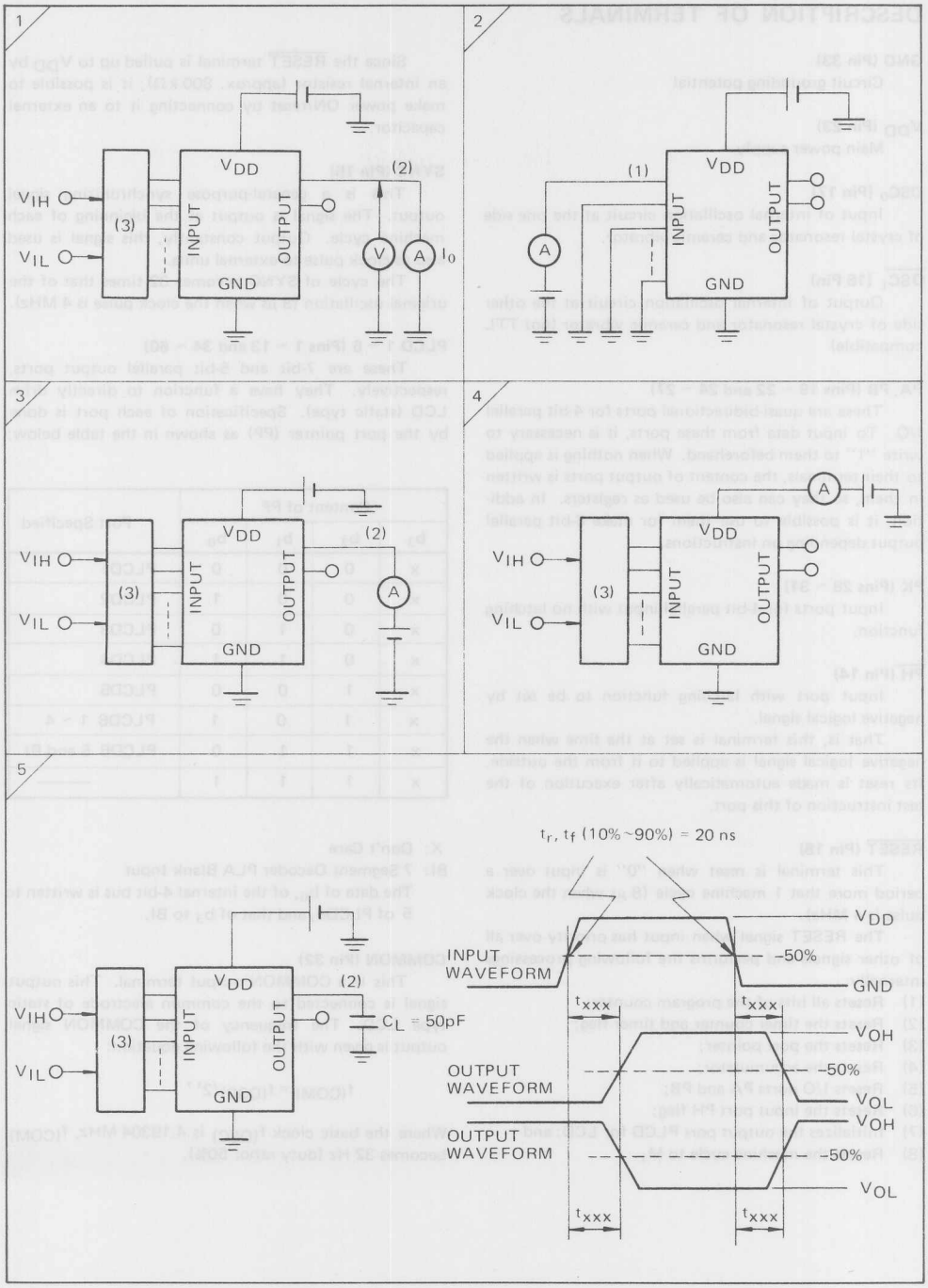




Notes:
(1) $t_{CY} = 32 \times 1/f_{(OSC)}$
(2) $t_{SW} = 4 \times 1/f_{(OSC)}$
(3) $t_{DV} = 8 \times 1/f_{(OSC)}$
(4) $t_{DIV} = 16 \times 1/f_{(OSC)} + 0.5\ \mu s$
(5) $t_{DDS} = 26 \times 1/f_{(OSC)} + 1\ \mu s$
(6) When data is input from PA or PB, set the content of PA or PB to "1" before it.
(7) At execution of the THB instruction, any input made during a period of T_{INH} $(15 \times 1/f_{(OSC)})$ shown in the above figure is neglected.

MEASURING CIRCUIT





Notes: (1) Repeated for the specified input terminal
(2) Repeated for the specified output terminal
(3) Input logic to place the system into the specified status

DESCRIPTION OF TERMINALS

GND (Pin 33)

Circuit grounding potential

V_{DD} (Pin 23)

Main power supply

OSC_0 (Pin 17)

Input of internal oscillation circuit at the one side of crystal resonator and ceramic vibrator.

$\overline{OSC_1}$ (16 Pin)

Output of internal oscillation circuit at the other side of crystal resonator and ceramic vibrator (not TTL compatible)

PA, PB (Pins 19 ~ 22 and 24 ~ 27)

These are quasi-bidirectional ports for 4-bit parallel I/O. To input data from these ports, it is necessary to write "1" to them beforehand. When nothing is applied to their terminals, the content of output ports is written in them, so they can also be used as registers. In addition, it is possible to use them for make 8-bit parallel output depending on instructions.

PK (Pins 28 ~ 31)

Input ports for 4-bit parallel input with no latching function.

$\overline{PH}$ (Pin 14)

Input port with latching function to be set by negative logical signal.

That is, this terminal is set at the time when the negative logical signal is applied to it from the outside. Its reset is made automatically after execution of the test instruction of this port.

$\overline{RESET}$ (Pin 18)

This terminal is reset when "0" is input over a period more that 1 machine cycle (8 μs when the clock pulse is 4 MHz).

The RESET signal when input has priority over all of other signals and performs the following processings internally:

(1) Resets all bits of the program counter;
(2) Resets the timer counter and timer flag;
(3) Resets the port pointer;
(4) Resets the accumulator;
(5) Resets I/O ports PA and PB;
(6) Resets the input port PH flag;
(7) Initializes the output port PLCD for LCD; and
(8) Resets the machine cycle to M_1.

Since the $\overline{RESET}$ terminal is pulled up to V_{DD} by an internal resistor (approx. 800 kΩ), it is possible to make power ON/reset by connecting it to an external capacitor.

SYNC (Pin 15)

This is a general-purpose synchronizing signal output. The signal is output at the biginning of each machine cycle. Output constantly, this signal is used also as clock pulse to external units.

The cycle of SYNC becomes 32 times that of the original oscillation (8 μs when the clock pulse is 4 MHz).

PLCD 1 ~ 6 (Pins 1 ~ 13 and 34 ~ 60)

These are 7-bit and 5-bit parallel output ports, respectively. They have a function to directly drive LCD (static type). Specification of each port is done by the port pointer (PP) as shown in the table below;

Content of PP				Port Specified
b_3	b_2	b_1	b_0	
x	0	0	0	PLCD1
x	0	0	1	PLCD2
x	0	1	0	PLCD3
x	0	1	1	PLCD4
x	1	0	0	PLCD5
x	1	0	1	PLCD6 1 ~ 4
x	1	1	0	PLCD6 5 and BI
x	1	1	1	———

X: Don't Care

BI: 7 Segment Decoder PLA Blank Input
The data of b_0, of the internal 4-bit bus is written to 5 of PLCD6, and that of b_3 to BI.

COMMON (Pin 32)

This is a COMMON output terminal. This output signal is connected to the common electrode of static type LCD. The frequency of the COMMON signal output is given with the following equation:

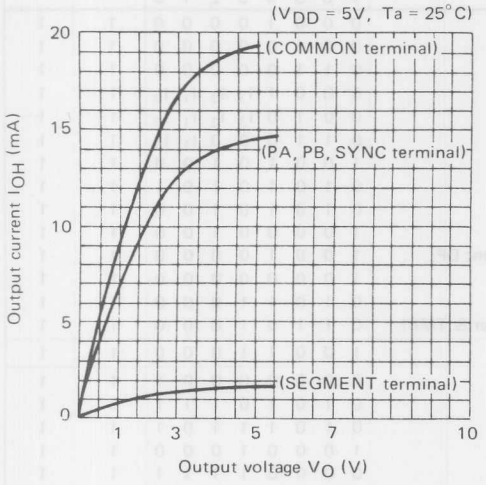
$$f_{(COM)} = f_{(OSC)}/2^{17}$$

Where the basic clock $f_{(OSC)}$ is 4.19304 MHz, $f_{(COM)}$ becomes 32 Hz (duty ratio: 50%).

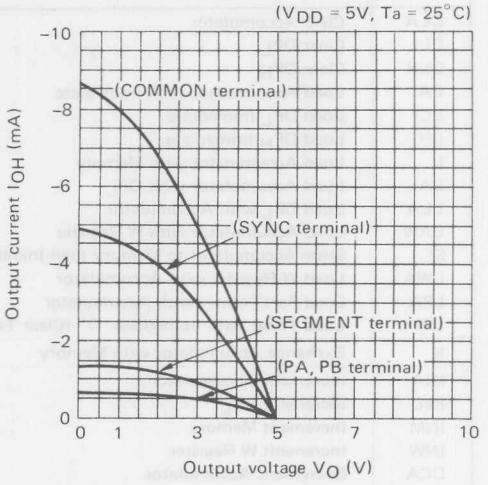
INSTRUCTIONS LIST

Mnemonic	Description	Instruction Code 7	6	5	4	3	2	1	0	Byte	Cycle
CLA	Clear Accumulator	0	0	0	1	0	0	0	0	1	1
CLL	Clear DP_L	0	0	1	0	0	0	0	0	1	1
CLH	Clear DP_H	0	1	1	0	0	0	0	0	1	1
LAI	Load Accumulator with Immediate	0	0	0	1	I_3	I_2	I_1	I_0	1	1
LLI	Load DP_L Immediate	0	0	1	0	I_3	I_2	I_1	I_0	1	1
LHI	Load DP_H Immediate	0	1	1	0	0	0	I_1	I_0	1	1
L	Load Accumulator with Memory	1	0	0	1	0	1	0	0	1	1
LAL	Load Accumulator with DP_L	0	1	0	1	0	1	0	1	1	1
LLA	Load DP_L with Accumulator	0	1	0	1	0	1	0	0	1	1
LAW	Load Accumulator with W Register	1	0	0	0	0	1	0	0	1	1
SI	Store Accumulator to Memory then Increment DP_L	1	0	0	1	0	0	0	0	1	1
LWA	Load W Register with Accumulator	1	0	0	0	0	0	0	0	1	1
LPA	Load Port Pointer with Accumulator	0	1	0	1	1	0	0	0	1	1
LTI	Load Timer with Immediate "0" (Clear Timer & TMF)	0	1	1	0	1	0	0	0	1	1
X	Exchange Accumulator with Memory	1	0	0	1	1	0	0	0	1	1
INA	Increment Accumulator	0	0	0	0	0	0	0	1	1	1
INL	Increment DP_L	0	1	0	1	0	1	1	1	1	1
INM	Increment Memory	0	1	0	1	1	1	0	1	1	1
INW	Increment W Register	1	0	0	0	1	0	0	0	1	1
DCA	Decrement Accumulator	0	0	0	0	1	1	1	1	1	1
DCL	Decrement DP_L	0	1	0	1	0	1	1	0	1	1
DCM	Decrement Memory	0	1	0	1	1	1	0	0	1	1
CAO	Complement Accumulator of One	0	1	0	1	0	0	0	0	1	1
RAL	Rotate Accumulator Left through Carry	0	1	0	0	0	1	1	1	1	1
AC	Add Memory to Accumulator with Carry	0	1	0	0	1	1	0	0	1	1
AS	Add Memory to Accumulator, Skip it Carry	0	1	0	0	1	1	1	0	1	1
AIS	Add Immediate to Accumulator, Skip if Carry	0	0	0	0	I_3	I_2	I_1	I_0	1	1
DAS	Decimal adjust Accumulator In Subtraction	0	1	0	1	1	0	1	0	1	1
CM	Compare Accumulator with Memory	0	1	0	1	1	1	1	0	1	1
SMB	Set Memory Bit	1	0	1	1	1	0	I_1	I_0	1	1
RMB	Reset Memory Bit	1	0	1	1	1	1	I_1	I_0	1	1
TAB	Test Accumulator Bit	1	0	1	0	0	0	I_1	I_0	1	1
TMB	Test Memory Bit	1	0	1	0	0	1	I_1	I_0	1	1
THB	Test H Port Bit	1	0	1	0	1	1	0	0	1	1
TTM	Test Timer flag	1	0	1	0	1	1	1	0	1	1
TC	Test Carry flag	0	1	0	0	0	0	1	0	1	1
SC	Set Carry flag	0	1	0	0	0	0	0	0	1	1
RC	Reset Carry flag	0	1	0	0	0	0	0	1	1	1
J	Jump	0	0	1	1	0	I_{10}	I_9	I_8	2	2
		I_7	I_6	I_5	I_4	I_3	I_2	I_1	I_0		
JC	Jump in Current Page	1	1	I_5	I_4	I_3	I_2	I_1	I_0	1	1
JA	Jump with Accumulator	0	1	0	0	0	0	1	1	1	1
CAL	Call Subroutine	0	0	1	1	1	I_{10}	I_9	I_8	2	2
		I_7	I_6	I_5	I_4	I_3	I_2	I_1	I_0		
RT	Return from Subroutine	0	1	0	1	1	0	0	1	1	1
OTD	Output Table Data	0	1	1	1	0	0	0	1	1~15	2
OA	Output Accumulator to Port A	0	1	1	1	0	0	1	0	1	1
OB	Output Accumulator to Port B	0	1	1	1	0	0	1	1	1	1
OP	Output Accumulator to Port designated Port Pointer	0	1	1	1	0	1	0	0	1	1
OPM	Output Memory to Port designated Port Pointer	0	1	1	1	0	1	1	0	1	1
IA	Input Port A in Accumulator	0	1	1	1	1	0	1	0	1	1
IB	Input Port B in Accumulator	0	1	1	1	1	0	1	1	1	1
IK	Input Port K in Accumulator	0	1	1	1	1	1	0	0	1	1
NOP	No Operation	0	0	0	0	0	0	0	0	1	1

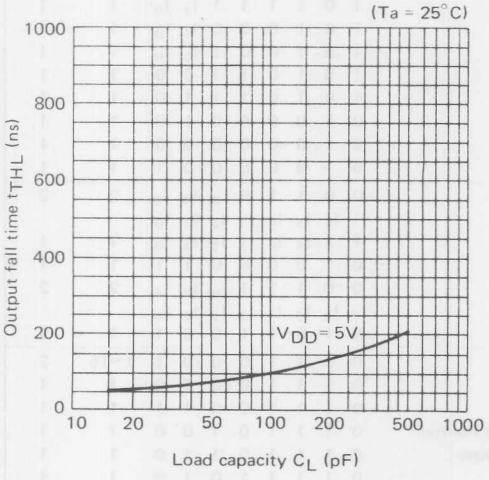
Output Current (I_{OL}) TYP



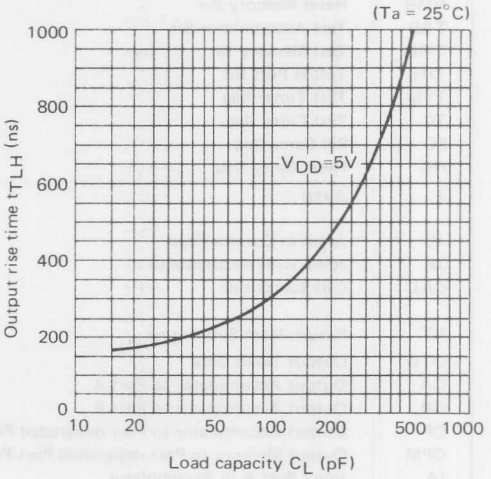
Output Current (I_{OH}) TYP



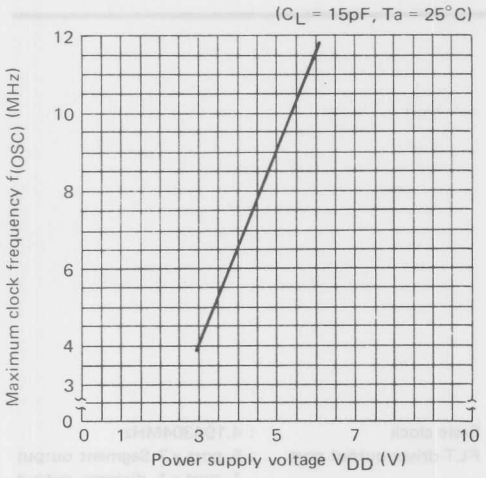
t_{THL} – C_L Characteristic TYP



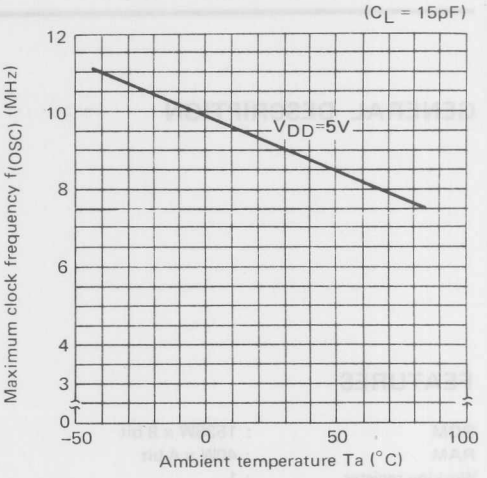
t_{TLH} – C_L Characteristic TYP



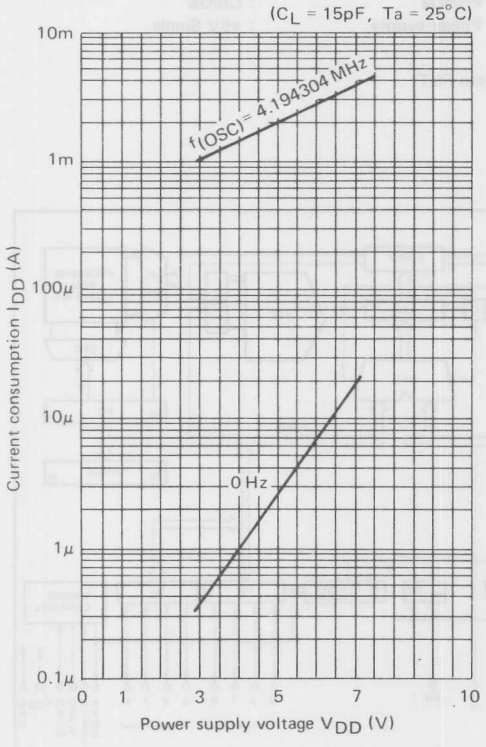
$f_{(OSC)}$ – V_{DD} Characteristic TYP



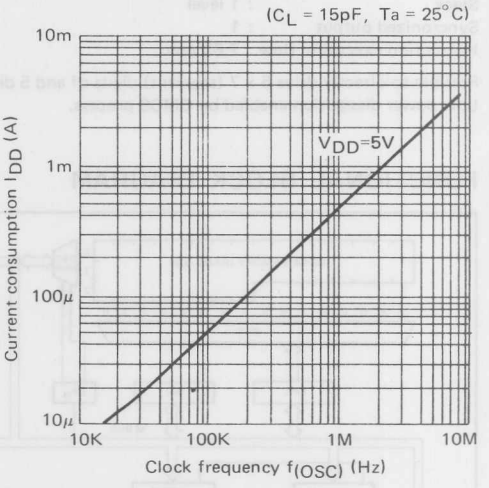
$f_{(OSC)}$ – Ta Characteristic TYP



I_{DD} – V_{DD} Characteristic TYP



I_{DD} – $f_{(OSC)}$ Characteristic TYP



OKI semiconductor

MSM58422GS

CMOS 4-BIT MICROCOMPUTER

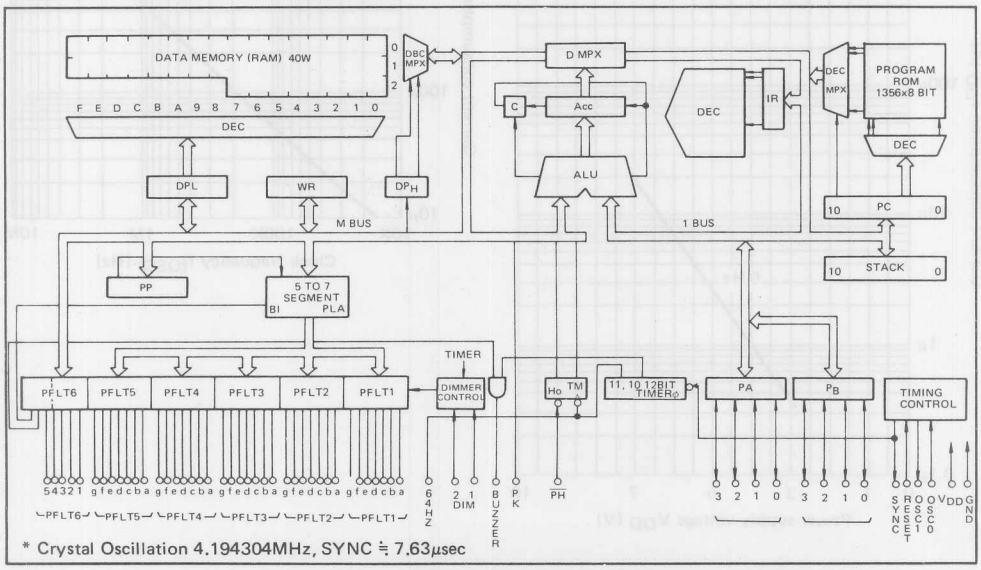
GENERAL DESCRIPTION

FEATURES

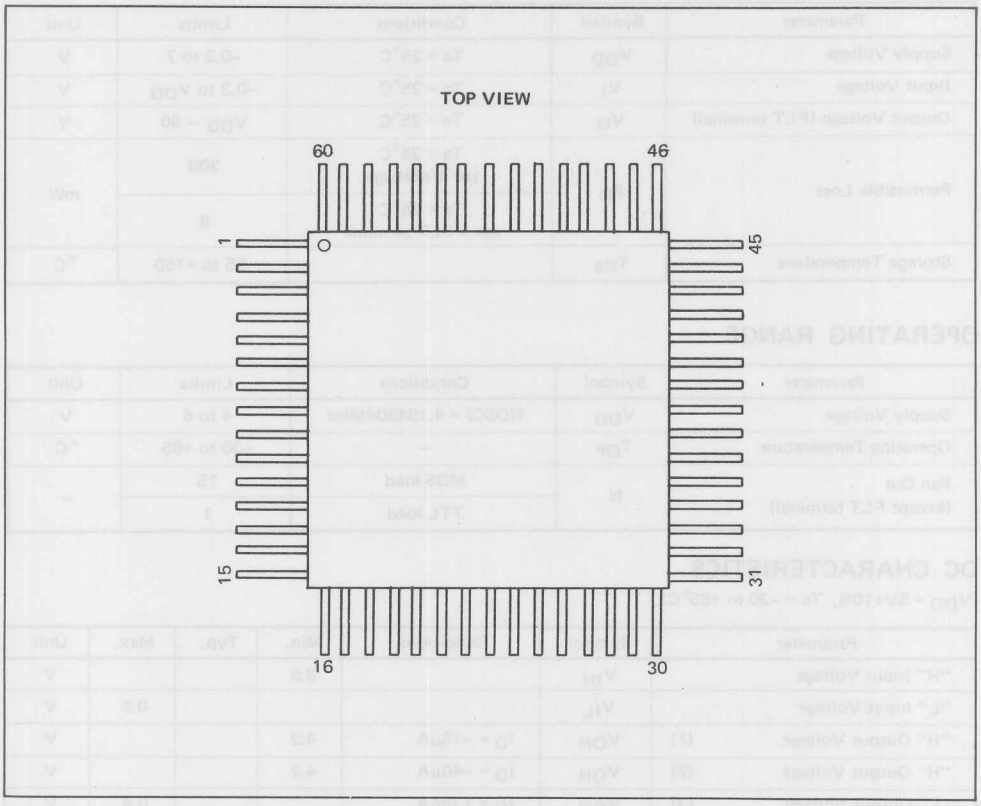
ROM	: 1536W x 8 bit	Basic clock	: 4.194304MHz
RAM	: 40W x 4 bit	FLT driver output port	: 5 port x 7 Segment output
Working register	: 1		1 port x 1 discrete output
Instructions	: 52		1 port x 4 discrete output
Input/Output port	: 8 (with output latch)	Dimmer input	: 2
Counter	: 11 bit	Package	: Plastic 60 pin flat
Stack	: 1 level	Process	: CMOS
Syncronized output	: 1	Power supply	: +5V Single
Instruction execution time	: 7.63 μsec		

Possible to directly drive 5 x 7 (segment) digits of and 5 discrete FLT.
Low power dissipation enabled by CMOS process.

FUNCTIONAL BLOCK DIAGRAM



PIN CONFIGURATION



TERMINAL NAME

Pin No.	Name	Pin No.	Name	Pin No.	Name	Pin No.	Name
1	PFLT5–c	16	$\overline{OSC1}$	31	PORT K	46	PFLT3–e
2	" 3–a	17	OSC0	32	64HZ	47	" 3–g
3	" 3–b	18	$\overline{RESET}$	33	GND	48	" 4–c
4	" 5–d	19	PORT A_0	34	PFLT6–1	49	" 4–d
5	" 2–f	20	" A_1	35	" 1–g	50	" 4–e
6	" 2–a	21	" A_2	36	" 1–c	51	" 4–g
7	" 5–e	22	" A_3	37	" 1–d	52	" 6–2
8	" 2–b	23	VDD	38	" 1–e	53	" 6–3
9	" 1–f	24	PORT B_0	39	" 2–g	54	" 6–4
10	" 5–f	25	" B_1	40	" 2–c	55	" 5–a
11	" 1–a	26	" B_2	41	" 2–d	56	" 4–f
12	" 1–b	27	" B_3	42	" 2–e	57	" 4–a
13	" 5–g	28	DIM 1	43	" 6–5	58	" 5–b
14	$\overline{PH}$	29	" 2	44	" 3–c	59	" 4–b
15	SYNC	30	BUZZER	45	" 3–d	60	" 3–f

ABSOLUTE MAXIMUM RATINGS

Parameter	Symbol	Conditions	Limits	Unit
Supply Voltage	V_{DD}	Ta = 25°C	–0.3 to 7	V
Input Voltage	V_I	Ta = 25°C	–0.3 to V_{DD}	V
Output Voltage (FLT terminal)	V_O	Ta = 25°C	V_{DD} – 30	V
Permissible Loss	P_D	Ta = 25°C per 1 package	200	mW
		Ta = 25°C per 1 FLT terminal	8	
Storage Temperature	T_{stg}		–55 to +150	°C

OPERATING RANGE

Parameter	Symbol	Conditions	Limits	Unit
Supply Voltage	V_{DD}	f(OSC) = 4.194304MHz	4 to 6	V
Operating Temperature	T_{OP}	–	–30 to +85	°C
Fan Out (except FLT terminal)	N	MOS load	15	–
		TTL load	1	

DC CHARACTERISTICS

(V_{DD} = 5V±10%, Ta = –30 to +85°C)

Parameter		Symbol	Conditions	Min.	Typ.	Max.	Unit
"H" Input Voltage		V_{IH}		3.6			V
"L" Input Voltage		V_{IL}				0.8	V
"H" Output Voltage	(1)	V_{OH}	I_O = –15μA	4.2			V
"H" Output Voltage	(2)	V_{OH}	I_O = –40μA	4.2			V
"L" Output Voltage	(3)	V_{OH}	I_O = 1.6mA			0.4	V
OSC_0 Input Current		I_{IH}/I_{IL}	V_I = V_{DD}/0V			10/–10	μA
Input Current	(4)	I_{IH}/I_{IL}	V_I = V_{DD}/0V			1/–20	μA
"H" Output Current	(1)	I_{OH}	V_O = 0.4V			–1	mA
"H" Output Current	(5)	I_{OH}	V_O = 2.5V	–0.25			mA
"H" Output Current	(6)	I_{OH}	V_O = 3V	–1			mA
"H" Output Current	(7)	I_{OH}	V_O = 3V	–1			mA
"L" Output Current	(8)	I_{OL}	V_O = 0.4V	1.6			mA
FLT Output Leak Current		I_{LO}	V_O = V_{DD}–26V			–10	μA
Input Capacitance		C_I	f=1MHz, Ta=25°C		5		pF
Output Capacitance		C_O	f=1MHz, Ta=25°C		7		pF
Supply Current		I_{DD}	f=4.194304MHz C_{IN}, C_{OUT}=30pF no load		2	5	mA

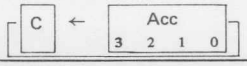
Notes: (1) PA, PB
(2) SYNC, BUZZER, 64Hz, FLT
(3) PA, PB, SYNC, BUZZER, 64Hz
(4) PH, RESET, DIM, PK
(5) SYNC, 64Hz
(6) FLT
(7) BUZZER

INSTRUCTION DESCRIPTION

Instruction Codes

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NOP	INA							AIS							DCA
1	CLA								LAI							
2	CLL								LLI							
3			J									CAL				
4	SC	RC	TC	JA				RAL					AC		AS	
5	CAO				LLA	LAL	DCL	INL	LPA	RT	DAS		DCM	INM	CM	
6	CLH	LHI							LTI							
7		OTD	OA	OB	OP		OPM				IA	IB	IK			
8	LWA				LAW				INW							
9	SI				L				X							
A		TAB				TMB							THB		TTM	
B										SMB				RMB		
C									JC							
D									JC							
E									JC							
F									JC							

Instruction Description

Type	Mnemonic	Instruction Code 7	6	5	4	3	2	1	0	Byte	Machine Cycle	Description
Clear, Load, Store and Exchange	CLA	0	0	0	1	0	0	0	0	1	1	Acc ←0
	CLL	0	0	1	0	0	0	0	0	1	1	DP_L ←0
	CLH	0	1	1	0	0	0	0	0	1	1	DP_H ←0
	LTI	0	1	1	0	1	0	0	0	1	1	$TC_{11\sim0}$ ←0, TM ←0
	LAI	0	0	0	1	I_3	I_2	I_1	I_0	1	1	Acc ←$I_3 \sim I_0$
	LLI	0	0	1	0	I_3	I_2	I_1	I_0	1	1	DP_L ←$I_3 \sim I_0$
	LHI	0	1	1	0	0	0	I_1	I_0	1	1	DP_H ←$I_1 \sim I_0$
	L	1	0	0	1	0	1	0	0	1	1	Acc ←(M)
	LAL	0	1	0	1	0	1	0	1	1	1	Acc ←DP_L
	LLA	0	1	0	1	0	1	0	0	1	1	DP_L ←Acc
	LAW	1	0	0	0	0	1	0	0	1	1	Aee ←W
	LWA	1	0	0	0	0	0	0	0	1	1	W ←Acc
	LPA	0	1	0	1	1	0	0	0	1	1	PP ←Acc
	SI	1	0	0	1	0	0	0	0	1	1	(M) ←Acc, DP_L ←DP_L 1, Skip if DP_L =0
	X	1	0	0	1	1	0	0	0	1	1	(M) ⟷Acc
Increment and Decrement	INA	0	0	0	0	0	0	0	1	1	1	Acc ←Acc + 1, Skip if Acc = 0
	INL	0	1	0	1	0	1	1	1	1	1	DP_L ←DP_L +1, Skip if DP_L = 0
	INM	0	1	0	1	1	1	0	1	1	1	(M) ←(M) + 1, Skip if (M) = 0
	INW	1	0	0	0	1	0	0	0	1	1	W ←W + 1, Skip if W = 0
	DCA	0	0	0	0	1	1	1	1	1	1	Acc ←Acc + F, Skip if Carry = 1
	DCL	0	1	0	1	0	1	1	0	1	1	DP_L ←DP_L−1, Skip if DP_L = F
	DCM	0	1	0	1	1	1	0	0	1	1	(M) ←(M) − 1, Skip if (M) = F
Arithmetic Operation	AC	0	1	0	0	1	1	0	0	1	1	C, Acc ←(M) + Acc + C
	AS	0	1	0	0	1	1	1	0	1	1	Acc ←(M) + Acc, Skip if Carry = 1
	AIS	0	0	0	0	I_3	I_2	I_1	I_0	1	1	Acc ←Acc + $I_{3\sim0}$, Skip if Carry = 1
	DAS	0	1	0	1	1	0	1	0	1	1	Acc ←Acc + 10 (Decimal)
Bit Operation	SMB	1	0	1	1	1	0	I_1	I_0	1	1	[(M), bit (I_1, I_0)] ←1
	RMB	1	0	1	1	1	1	I_1	I_0	1	1	[(M) bit (I_1, I_0)] ←0
	TMB	1	0	1	0	0	1	I_1	I_0	1	1	Skip if [(M), bit (I_1, I_0)] = 1
	THB	1	0	1	0	1	1	0	0	1	1	Skip if H = 1, H ← 0
	TTM	1	0	1	0	1	1	1	0	1	1	Skip if TM = 1, TM ←0
	TC	0	1	0	0	0	0	1	0	1	1	Skip if C = 1
	TAB	1	0	1	0	0	0	I_1	I_0	1	1	Skip if [Acc, bit (I_1, I_0)] = 1
	SC	0	1	0	0	0	0	0	0	1	1	C ←1
	RC	0	1	0	0	0	0	0	1	1	1	C ←0
Comparison and Shift	CM	0	1	0	1	1	1	1	1	1	1	Skip if (M) = Acc
	CAO	0	1	0	1	0	0	0	0	1	1	Acc ←$\overline{Acc}$
	RAL	0	1	0	0	0	1	1	1	1	1	

Type	Mnemonic	Instruction Code 7	6	5	4	3	2	1	0	Byte	Machine Cycle	Description
Jump, Call and Return	J	0	0	1	1	0	I_{10}	I_9	I_8	2	2	$PC_{10\sim0} \leftarrow I_{10\sim0}$
		I_7	I_6	I_5	I_4	I_3	I_2	I_1	I_0			
	JC	1	1	I_5	I_4	I_3	I_2	I_1	I_0	1	1	$PC_{5\sim0} \leftarrow I_{5\sim0}$
	JA	0	1	0	0	0	0	0	1	1	1	$PC_{3\sim0} \leftarrow$ Acc
	CAL	0	0	1	1	1	I_{10}	I_9	I_8	2	2	$STACK_{10\sim0} \leftarrow PC + 2$, $PC_{10\sim0} \leftarrow I_{10\sim0}$
		I_7	I_6	I_5	I_4	I_3	I_2	I_1	I_0			
	RT	0	1	0	1	1	0	0	1	1	1	$PC_{10\sim0} \leftarrow STACK_{10\sim0}$
Output	OTD	0	1	1	1	0	0	0	1	1~15	2	PA, PB $\leftarrow I_{7\sim0}$ (One Byte of Table Data)
		I_7	I_6	I_5	I_4	I_3	I_2	I_1	I_0			Table Data (MAX15 byte)
		I_7	I_6	I_5	I_4	I_3	I_2	I_1	I_0			
	OA	0	1	1	1	0	0	1	1	1	1	PA $\leftarrow$ Acc
	OB	0	1	1	1	0	0	1	1	1	1	PB $\leftarrow$ Acc
	OP	0	1	1	1	0	1	0	0	1	1	(P) $\leftarrow$ Acc
	OPM	0	1	1	1	0	1	1	1	1	1	(P) $\leftarrow$ (M)
Input	IA	0	1	1	1	1	0	1	1	1	1	Acc $\leftarrow$ PA
	IB	0	1	1	1	1	0	1	1	1	1	Acc $\leftarrow$ PB
	IK	0	1	1	1	1	1	0	0	1	1	Acc $\leftarrow$ PK (Acc 2~0 = "1")
Other	NOP	0	0	0	0	0	0	0	0	1	1	No Operation

Preliminary

MSM6502

CMOS 4-BIT SINGLE CHIP MICROCOMPUTER

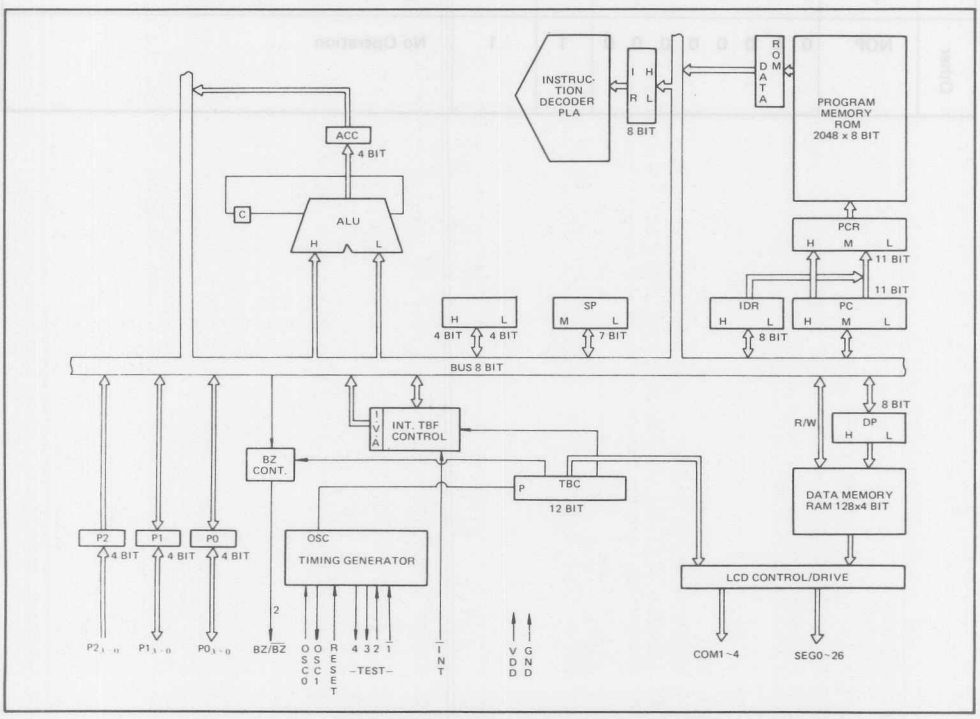
GENERAL DESCRIPTION

The MSM6502 is a low power high performance 4-bit single chip microcomputer using the CMOS silicon gate process. It has the built-in LCD driving circuit.

FEATURES

ROM	: 2,048 words x 8 bits	Power Down	: INT and I/O release with the HALT instruction
RAM	: 128 words x 4 bits	Buzzer Output	: 2K/1K/512 Hz program option, complement output
I/O Port	: 2 x 4 bits	Time Base Flag	: 2 (128 Hz and 16 Hz)
Input Port	: 1 x 4 bits	Supply Voltage	: 2.4V ~ 3.6V
LCD Segment	: 108 (1/4 duty)	Power Consumption	: 30 μA typically (10 μA typically at the halt mode)
Clock Oscillation	: 32.768 kHz crystal oscillation	Package	: 56-pin plastic flat package
Cycle Time	: 91.5 μsec.	Bias for LCD Drive	: Internally generated
Interruption	: 3-level		
Stack	: SP = 7, Nesting RAM		
Instruction Set	: 68		

FUNCTIONAL BLOCK DIAGRAM



LIST OF PINS

Pin No.	Description	Pin No.	Description	Pin No.	Description	Pin No.	Description
1	P00	15	$\overline{\text{RESET}}$	29	SEG0	43	SEG14
2	P01	16	$\overline{\text{INT}}$	30	SEG1	44	SEG15
3	P02	17	$\overline{\text{TEST1}}$	31	SEG2	45	SEG16
4	P03	18	$\overline{\text{TEST2}}$	32	SEG3	46	SEG17
5	P10	19	TEST3	33	SEG4	47	SEG18
6	P11	20	TEST4	34	SEG5	48	SEG19
7	P12	21	VDD	35	SEG6	49	VDD
8	P13	22	OSC0	36	SEG7	50	SEG20
9	P20	23	$\overline{\text{OSC1}}$	37	SEG8	51	SEG21
10	P21	24	GND	38	SEG9	52	SEG22
11	P22	25	COM1	39	SEG10	53	SEG23
12	P23	26	COM2	40	SEG11	54	SEG24
13	BZ	27	COM3	41	SEG12	55	SEG25
14	$\overline{\text{BZ}}$	28	COM4	42	SEG13	56	SEG26

PIN DESCRIPTION

I/O Ports 0 and 1 (P0 and P1)

P0 and P1 are pseudo bidirectional ports for 4-bit parallel data input and output. A bit operation of the set, reset and test (SPB, RPB and TPB) is also possible.

The port is specified by the L register internally.

To input data from these ports, 1's must be written by the output instructions, or the IP (input) or TPB instruction must be executed. 1's are output in the reset mode.

P03 is also used to release the halt mode. If the HALT instruction is executed when P03 is 1, the halt mode is released as soon as P03 becomes 0, and vice versa.

Input Ports 2 (P2)

P2 are for 4-bit data parallel input. Each bit can also be tested by the TPB instruction.

The input circuits of P2 are so designed as to minimize the power consumption. They are pulled down with 5K ohms at no instruction execution, while they are pulled up with 130K ohms at the instruction execution (IP or TPB).

The input from the external circuit must be 0 or open. The open signal input is recognized as 1 by the CPU.

Buzzer Pins (BZ and $\overline{\text{BZ}}$)

BZ and $\overline{\text{BZ}}$ are used to generate the sound like alarm. The output frequency can be selected from 512, 1024 and 2048 Hz by the program. The selection is made at P3.

Clock Oscillation Pins (OSC0 and $\overline{\text{OSC1}}$)

By connecting the 32.768 kHz crystal and the condenser to these pins, the basic clock pulse to operate the MSM6502 is obtained.

Test Pins ($\overline{\text{TEST1}}$ ~ TEST4)

Test pins must be normally open.

Interrupt Pin ($\overline{\text{INT}}$)

This pin is used to request an interrupt from the external circuit. The internal flag is set by the fall of the input signal. For detail procedure of interrupt operation, refer to the 6-11).

Reset Pin ($\overline{\text{RESET}}$)

By the input of 0 to this pin, the MSM6502 is placed in the initial mode. As this pin is provided with the built-in pull up resistance, the power-on-reset can be done by connecting the condenser to this pin externally.

The reset signal has a priority to all other signals and performs following operations automatically.

- Resets all bits of the PC to 0.
- Sets all bits of P1 and P2 to 1.
- Resets the internal registers (H, L, ACC and C)
- Resets the skip flag.
- Resets the time base counter (TBC).
- Resets the interrupt request flag (IRQF).
- Resets the interrupt enable flag (EIF). (DI mode)
- Resets the master interrupt enable flag (MEIF). (DI mode)
- Sets all bits of the stack pointer (SP).
- Initializes the segment and common outputs.

The reset operation is made by the 2-machine cycle instruction. The reset mode is released after 1 is input to the $\overline{\text{RESET}}$ pin and the TBC starts operation.

LCD Drive Pins (COM1 ~ 4 and SEG0 ~ 26)

The MSM6502 can drive the LCD with 1/4 duty and 1/3 bias. The maximum number of segments is 4 x 27 = 108.

COM1 ~ 4 are connected to the common electrode and the SEG0 ~ 26 are connected to the segment electrode. The same electric potential is output at COM1 ~ 4 and SEG0 ~ 26 in the reset mode, so that the DC bias is not applied to the LCD even if the CPU's operation is being stopped (the interval between the power on and the CPU's starting operation). Even if the CPU starts operation, the LCD is blank unless the display enable flag is set by the instruction. COM1 ~ 4 and SEG0 ~ 26 output the no display waveform at that time.

Supply Voltage Pin (VDD and GND)

+3V is supplied to VDD typically and 0V is supplied to GND.

FUNCTIONAL DESCRIPTION

Programmable ROM

The programmable ROM has a capacity of 2,048 bytes x 8 bits. 2,000 bytes (000 ~ 7CFH) are used for the user's program, while 48 bytes (7DOH ~ 7FFH) are used for the test program. The ROM is divided into 32 pages x 64 addresses/page.

The address 0 of the page 0 is the initial address after reset operation. The address 16 (010H) and the address 24 (018H) of the page 0 are used for the internal interrupt, while the address 32 (020H) of the page 0 is used for the external interrupt.

The addresses 2,000 ~ 2,047 (7DOH ~ 7FFH) of the page 31 are used for the test program, so those addresses cannot be used for the user's program.

Program Counter (PC)

The PC is an 11-bit binary counter incremented by one at each instruction execution. However, it is not incremented at execution of such instructions as JC, CAL, RT and other 2-byte instructions. PC bits 10 ~ 6 specify the page and bits 5 ~ 0 specify the address in the specified page.

Index Register (IDR)

The IDR is an 8-bit register which specifies the eight low-order bits of the ROM address with the LMT instruction. The table data conversion and the segment conversion can be made without changing the PC, if the initial address of the data etc. have been stored in the IDR. The IDR's three high-order bits are equal to the contents of the PC.

Instruction Register (IR) & Instruction Decoder (PLA)

The IR is an 8-bit register which retains the instruction fetched from the ROM until the instruction execution is completed. The PLA is a logical circuit which reads the instruction retained in the IR and outputs the control signals necessary for the instruction execution.

Arithmetic Logic Unit (ALU)

The ALU is a logical circuit for performing the operation for the data code's four high-order bits and four low-order bits, in the internal bus.

It performs such operation as the arithmetic operation, the logical operation, the comparison and the rotate operation. The ALU operation result is stored in the ACC (accumulator), the TR (temporary register) and the C (carry flag).

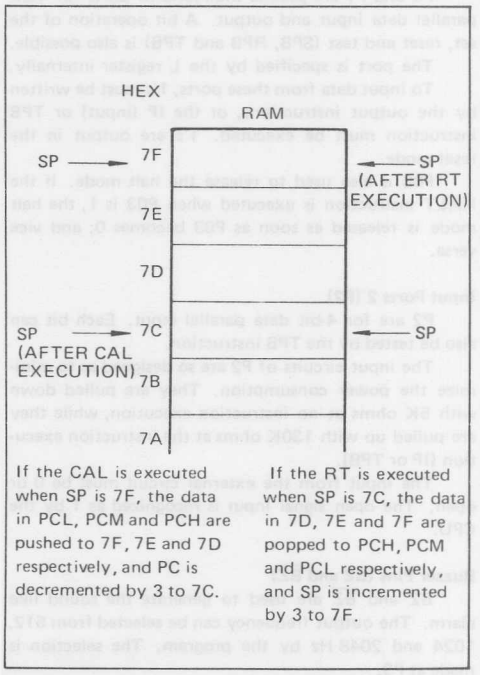
Registers H & L

The registers H and L are the 4-bit registers. The 8 bits of the combination of two registers specify the address of the data memory. (The one high-order bit is not effective). Each register can transfer the data with the ACC, so they can be used as the working register. The register L can also specify the port.

Stack, Stack Pointer (SP)

The data memory (RAM) is used to save the PC content and the register when a subroutine is called or an interrupt occurs. As the RAM's one word is composed of 4 bits, 3 words are necessary to save the PC contents and 2 words are necessary to save the register.

The SP is a 7-bit register to indicate the stack's initial address in the RAM. It is decremented when it is pushed and is incremented when it is popped.



columns x 8 rows. A word is selected by specifying the column with the four low-order bits of the DP (data pointer) and the row with the three high-order bits of the DP.

The DP is a 7-bit register operated by the hardware logic. The RAM is accessed by fetching the contents of the registers H and L or the second byte of the instruction to the DP.

The 27 words (00 ~ 1AH) are exclusively used as the display RAM to store the data for display. The RAM is also used as the stack. The contents of the display RAM is automatically transferred to the driver circuit.

The relations between the display RAM and the SEG/COM pins are as follows.

Display RAM	SEG pin	Display RAM	COM pin
00	0	Bit 0	1
01	1	1	2
02	2	2	3
13	3	3	4
⋮	⋮		
OF	15		
10	16		
⋮	⋮		
1AH	26		

LCD Controller & Driver

The MSM6502 can directly drive the LCD with 1/4 duty and 1/3 bias. The bias electric potential is obtained from the VDD by the resistance division. The common and the segment drivers are operated by the hardware logic, so they continuously operate as far as the CPU supply the clock.

Time Base Counter (TBC)

The TBC is a 12-bit binary counter which prescales 32.768 kHz to control the LCD driver, the time base flag and the buzzer.

The TBC always operates unless the MSM6502 is in reset mode, so the application program with the low power counting function can be developed by using the time base flag to release the HALT instruction.

flag) are prepared for each interrupt to control an interrupt. The MEIF (master enable flag) enables an interrupt itself.

An interrupt is requested as follows.

The fall of the INT input . . . . . External interrupt
The fall of the TBC's 128 Hz output
. . . . . Internal interrupt
The fall of the TBC's 16 Hz output
. . . . . Internal interrupt

When an interrupt is requested, the corresponding IRQF is set. If both of the MEIF and the corresponding INTEF are being set at that time, an interrupt operation is performed in the following order.

a. The MEIF is reset.
b. The PC content is pushed to the stack.
c. The address corresponding to the IRQF is loaded into the PC, and the IRQF accepted an interrupt is reset.

3-machine cycle is required for an interrupt operation. The load address to the PC is as follows.

External interrupt: 020H
Internal interrupt (16 Hz): 010H
Internal interrupt (128 Hz): 018H

Each IRQF can be tested by the TIRB instruction. The relation between the IRQF and the bits are as follows.

External interrupt IRQF : Bit 3
Internal interrupt (128 Hz) IRQF : Bit 2
Internal interrupt (16 Hz) IRQF : Bit 1

Port Register P3 & P4

P3 and P4 are port registers specified by the L register. The P3 is used to select the frequency for the buzzer output. The P4 is used as the LCD driver controller and the enable flag for each interrupt.

P3

Bit 3: The MEIF content is read by the IP instruction execution.
Bit 2: On(1)/Off(0) of 512 Hz output to BZ and $\overline{BZ}$.
Bit 1: On(1)/Off(0) of 1024 Hz output to BZ and $\overline{BZ}$.
Bit 0: On(1)/Off(0) of 2048 Hz output to BZ and $\overline{BZ}$.

P4

Bit 3: "1" enables an external interrupt.
Bit 2: "1" enables an internal 128 Hz interrupt.
Bit 1: "1" enables an internal 16 Hz interrupt.
Bit 0: On(1)/Off(0) of the LCD driver.

Timing Generator (TG)

The TG generates the basic clock pulse and the timing pulse necessary for the various control.

INSTRUCTION LIST

H \ L	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NOP								AIS							
1	IN A	IN L	IN M		DC A	DC L	DC M		RAL	RAR	LAMD <B2>	LMAD <B2>	XAMD <B2>		RC	SC
2	SPB				RPB				TPB				TAB			
3	SMB				RMB				TMB				RT	RTS		XHS
4									TIRB				AND	OR	TC	HALT
5	SMBD <B2>				RMBD <B2>				TMBD <B2>				INX	DCX	EI	DI
6	DSC	DAC	ADS	ADCS		CMA	EOR	LMT		LXI <B2>	LHLI <B2>	CAM	CPAL	CAXL	CAXM	
7	LLI															
8	LAI															
9	J				<B2>											
A	CAL				<B2>											
B	LAM	LAL	LAH	IP	LMA	LLA	LHA	OP	XAM				PUSH HL	PUSH CA	POP HL	POP CA
C	JCP															
D	JCP															
E	JCP															
F	JCP															

DESCRIPTION OF INSTRUCTIONS

	Mnemonic	Machine Language (H)	Function	Description
Logic/Arithmetic Operation	DAC	61	C, ACC ← C + ACC + M + 6, Adjust if C = 0	Adds the data memory contents addressed by the H and L, 6, the ACC contents and the C contents, and forms the result in the ACC and C. If the C contents are 0, adds the ACC contents and 10, and forms the result in the ACC. Decimal addition.
	DSC	60	C, ACC ← C + ACC + M, Adjust if C = 0	Performs a 1's complement operation on the data memory contents addressed by the H and L, adds the result and the ACC and C contents, and forms the result in the ACC and C. If the C contents are 0, adds the ACC contents and 10, and forms the result in the ACC. Decimal subtraction.
	ADS	62	Acc ← ACC + M, SKIP 15 Carry = 1	Adds the data memory contents addressed by the H and L and the ACC contents, and forms the result in the ACC. If a carry is generated, the next instruction is skipped.
	ADCS	63	C, ACC ← C + ACC + M, SKIP if C = 1	Adds the data memory contents addressed by the H and L and the ACC and C contents, and forms the result in the ACC and C. If the C contents are 1, the next instruction is skipped.
	AIS n	00~0F	ACC ← ACC + n, SKIP if Carry = 1	Adds the ACC contents and the four low-order bits of the instruction code bits, and forms the result in the ACC. If a carry is generated, the next instruction is skipped.
	EQR	66	ACC ← ACC ∀ M	Gets EOR (OR, AND) of the data memory contents addressed by the H and L and the ACC contents, and forms the result in the ACC.
	AND	4C	ACC ← ACC ∧ M	
	OR	4D	ACC ← ACC ∨ M	
	CMA	65	ACC ← $\overline{ACC}$	Performs 1's complement operation on the ACC contents and forms the result in the ACC.
Transfer	LHLI n8	6A 00~FF	H, L ← <B2>	Loads the four high-order bits and four low-order bits of the instruction code byte 2 into the H and L respectively.
	LXI n8	69 00~FF	X ← <B2>	Loads the four high-order bits and four low-order bits of the instruction code byte 2 into the IDR's four high-order bits and four low-order bits respectively.
	LAI n	80~8F	ACC ← n	Loads the four low-order bits of the instruction code bits into the ACC.
	LLI n	70~7F	L ← n	Loads the four low-order bits of the instruction code bits into the L.
	LAL	B1	ACC ← L	Loads the L contents into the ACC.
	LAH	B2	ACC ← H	Loads the H contents into the ACC.
	IP	B3	ACC ← P	Loads the contents of the port specified by the L into the ACC.

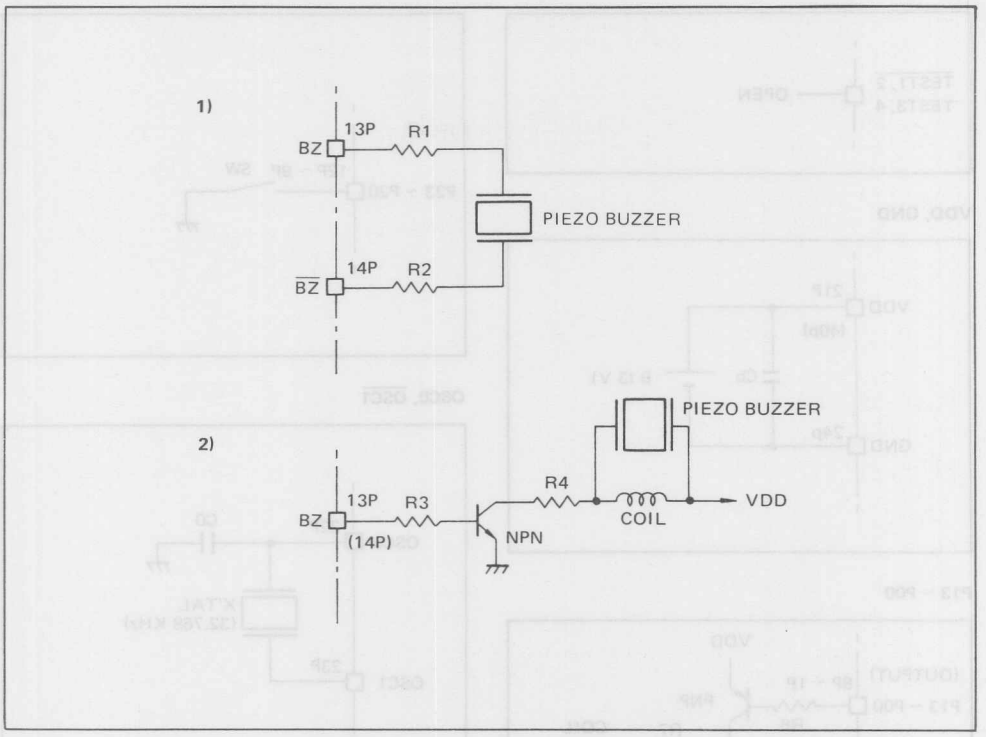
	Mnemonic	Machine Language (H)	Function	Description
Transfer	LLA	B5	L ← ACC	Loads the ACC contents into the L.
	LHA	B6	H ← ACC	Loads the ACC contents into the H.
	OP	B7	P ← ACC	Loads the ACC contents into the port specified by the L.
	LAM	B0	ACC ← M	Loads the data memory contents addressed by the H and L into the ACC.
	LMA	B4	M ← ACC	Loads the ACC contents into the data memory area addressed by the H and L.
	LMT	67	M <HL> ← ROM3~0 <HLH> 7~4	Loads the four high-order bits and four low-order bits of the ROM contents addressed by the PC's three high-order bits and the IDR's eight bits into the data memory area addressed by the (H, L) +1 and (H, L) respectively.
	LAMD m7	1A 00~7F	ACC ← M <m7>	Loads the data memory contents specified by the instruction code byte 2 into the ACC.
Transfer/Exchange	LMAD m7	1B 00~7F	M <m7> ← ACC	Loads the ACC contents into the data memory area specified by the instruction code byte 2.
	XAM	B8	M ⟷ ACC	Exchanges the data memory contents addressed by the H and L with the ACC contents.
	XAMD m7	IC 00~7F	M <m7> ⟷ ACC	Exchanges the data memory contents addressed by the H and L with the ACC contents.
	XHS	3F	H, L ⟷ SP	Exchanges the SP's four low-order bits with the L contents, and exchanges the SP's three high-order bits with the H contents. One high-order bit of the H remains unchanged.
Increment/Decrement	INA	10	ACC ← ACC + 1, SKIP if ACC = 0	Increments the ACC contents by one and forms result in the ACC. If the ACC contents go to 0, the next instruction is skipped.
	INL	11	L ← L + 1, SKIP if L = 0	Increments the L contents by one and forms the result in the L. If the L contents go to 0, the next instruction is skipped.
	INM	12	M ← M + 1, SKIP if M = 0	Increments the data memory contents addressed by the H and L by one and forms the result in that data memory area. If that data memory contents go to 0, the next instruction is skipped.
	INX	5C	X ← X + 1	Increments IDR contents by one and forms the result in the IDR.
	DCA	14	ACC ← ACC-1, SKIP if ACC = F	Decrements the ACC contents by one and forms the result in the ACC. If the ACC contents go to F, the next instruction is skipped.
	DCL	15	L ← L-1, SKIP if L = F	Decrements the L contents by one and forms the result in the L. If the L contents go to F, the next instruction is skipped.

	Mnemonic	Machine Language (H)	Function	Description
Increment/Decrement	DCM	16	M ← M-1, SKIP if M = F	Decrements the data memory contents addressed by the H and L and forms the result in that data memory area. If that data memory contents go to F, and next instruction is skipped.
	DCX	5D	X ← X – 1	Decrements the IDR contents by one and forms the result in the IDR.
Comparison	CAM	6B	SKIP if ACC = M	If the data memory contents addressed by the H and L and the ACC contents are equal, the next instruction is skipped.
	CPAL	6C	SKIP if ACC = L	If the L contents and the ACC contents are equal, the next instruction is skipped.
	CAXL	6D	SKIP if ACC = XL	If the IDR's four low-order bits and the ACC contents are equal, the next instruction is skipped.
	CAXH	6E	SKIP if ACC = XH	If the IDR's four high-order bits and the ACC contents are equal, the next instruction is skipped.
Bit Process	SMB n2	30~33	[M bit n] ← 1	Sets the data memory bit addressed by the H and L. The set bit is specified with the two low-order instruction code bits.
	SPB n2	20~23	[P bit n] ← 1	Sets the port bit addressed by the L. The set bit is specified with the two low-order instruction code bits.
	SMBD m7, n2	50~53 00~7F	[M<m7> bit n] ← 1	Sets the data memory bit addressed by the seven bits of the instruction code byte 2. The set bit is specified with the two low-order bits of the instruction code byte 1.
	5C	1F	C ← 1	Sets the C.
	RMB n2	34~37	[M bit n] ←0	Resets the data memory bit addressed by the H and L. The reset bit is specified with the two low-order instruction code bits.
	RPB n2	24~27	[P bit n] ← 0	Resets the port bit addressed by the L. The reset bit is specified with the two low-order instruction code bits.
	RMBD m7, n2	54~57 00~7F	[M<m7> bit n] ← 0	Resets the data memory bit addressed by the seven bits of the instruction code byte 2. The reset bit is specified with the two low-order bits of the instruction code byte 1.
	RC	1E	C ← 0	Resets the C.
	TMB n2	38~3B	SKIP if [M bit n] = 1	If the data memory bit specified with the two low-order instruction code bits is 1, the next instruction is skipped. The data memory is addressed by the H and L.
	TPB n2	28~2B	SKIP if [P bit n] = 1	If the port bit specified with the two low-order instruction code bits is 1, the next instruction is skipped. The prot is addressed by the L.

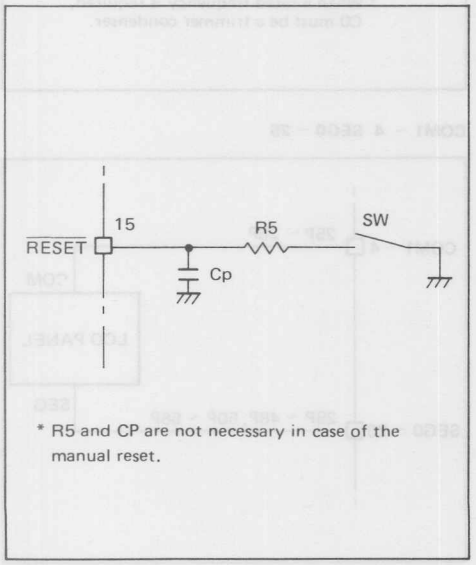
	Mnemonic	Machine Language (H)	Function	Description
Bit Process	TMBD m7, n2	58~5B 00~7F	SKIP if [M<m7> bit n] = 1	If the data memory bit specified with the low-order bits of the instruction code byte 1 is 1, the next instruction is skipped. The data memory is addressed by the seven bits of the instruction code byte 2.
	TC	4E	SKIP if C = 1	If the C contents are 1, the next instruction is skipped.
	TAB n2	2C~2F	SKIP if [ACC bit n] = 1	If the ACC bits specified with the two low-order instruction code bits is 1, the next instruction is skipped.
	TIRB n2	49~4B	SKIP if [IRQF bit n] = 1, then Reset	If the IRQF specified with the two low-order instruction code bits is 1, the next instruction is skipped and that IRQF is reset.
	RAL	18	┌C ← 3 ← 2 ← 1 ← 0┐	Shifts the ACC and C contents one bit to the left.
	RAR	19	ACC └C → 3 → 2 → 1 → 0┘	Shifts the ACC and C contents one bit to the right.
	EI	5E	MEIF ← "1"	Enables an interrupt.
	DI	5F	MEIF ← "0"	Disables an interrupt.
	HALT	4F	HMF ← "1"	Halts the operation. The clock oscillator, time base counter and LCD driver normally operate, however.
Branch	CAL n11	A0~A7 00~FF	STACK ← PC+2, PC ← n11 SP ← SP-3	Increments the PC contents by two and saves that in the stack register. Then loads the three low-order bits of the instruction code byte 1 and the eight bits of the instruction code byte 2 into the PC. The SP is decremented by three.
	J n11	90~97 00~FF	PC ← n11	Loads the three low-order bits of the instruction code byte 1 and the eight bits of the instruction code byte 2 into the PC.
	JCP n6	C0~FF	PC5 ~ 0 ← n6	Loads the six low-order instruction code bits into the six low-order PC bits.
	RT	3C	PC ← STACK, SP ← SP+3	Loads the stack contents into the PC. The SP is incremented by three.
	RTS	3D	PC ← STACK, SP ← SP+3	Loads the stack contents into the PC. The SP is incremented by three. The instruction addressed by the PC is skipped.
Push/Pop	PUSH HL	BC	STACK ← H, L, SP ← SP-2	Saves the H and L contents in the stack. The SP is decremented by two.
	PUSH CA	BD	STACK ← C, ACC, SP ← SP-2	Saves the C and ACC contents in the stack. The SP is decremented by two.
	POPHL	BE	H, L ← STACK, SP ← SP+2	Loads the stack contents into the H and L. The SP is incremented by two.
	POPCA	BF	C, ACC ← STACK, SP ← SP+2	Loads the stack contents into the C and ACC. The SP is incremented by two.
Others	NOP	00	PC ← PC+1	No operation. The PC is incremented by one.

PIN CONNECTIONS (SAMPLE)

BZ, $\overline{BZ}$

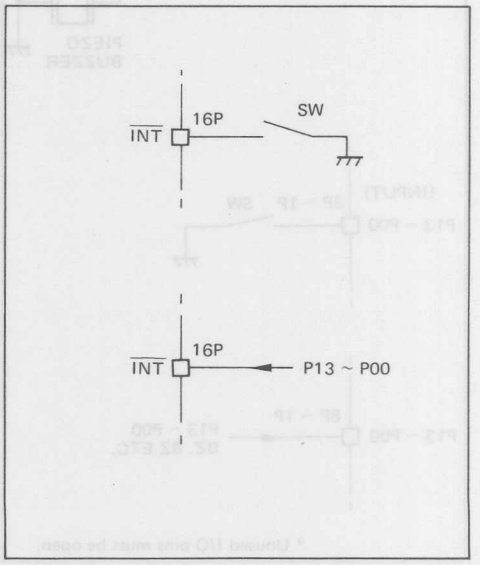


$\overline{RESET}$

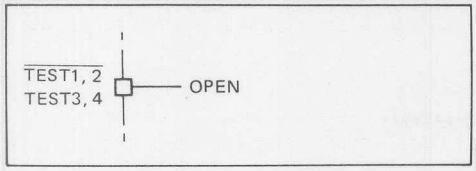


* R5 and CP are not necessary in case of the manual reset.

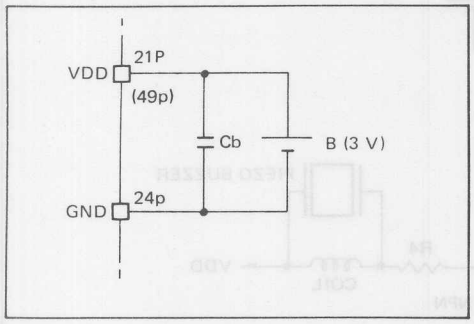
$\overline{INT}$



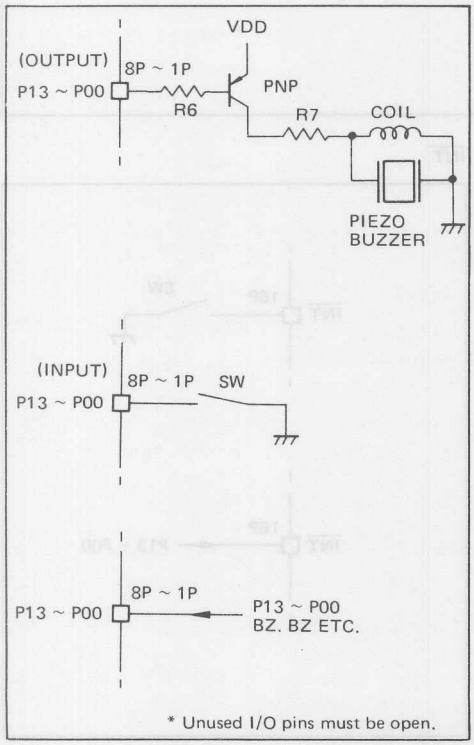
$\overline{\text{TEST 1, 2}}$, TEST 3,4



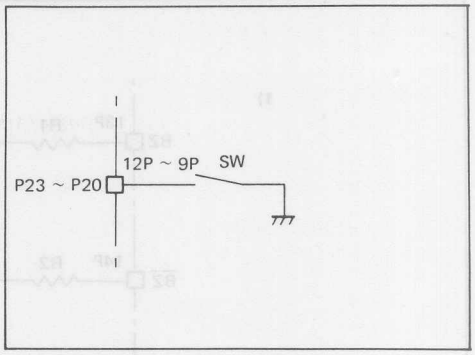
VDD, GND



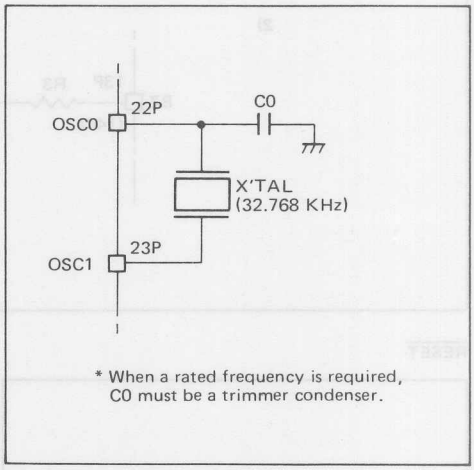
P13 ~ P00



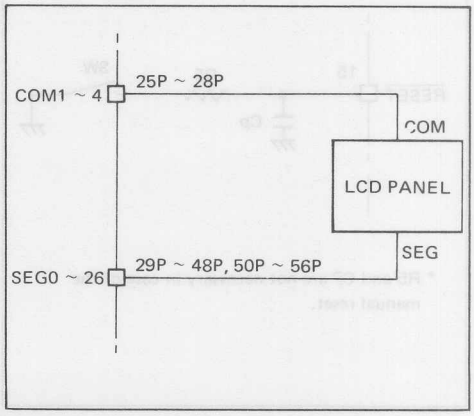
P23 ~ P20



OSC0, $\overline{\text{OSC1}}$



COM1 ~ 4 SEG0 ~ 26



FOREWORD

This manual deals with the hardware and software of the OLMS-64 Series of CMOS 4-bit single-chip microcomputers.

For explanation, this manual describes the MSM6404RS, the highest version of the series. When using a lower version chip, the reader is advised to utilize this manual by paying good attention to the differences in memory capacity, built-in functions, etc.

This manual is subject to revision without notice due to improvements of the product.

CHAPTER 1 GENERAL DESCRIPTION

1.1 OLMS-64 SERIES

Since the release of the OLMS-40 Series, the industry's first CMOS 4-bit single-chip microcomputer, in 1978, Oki Electric Industry has been expanding the series with full utilization of CMOS technology.

The OLMS-40 Series, used in a large variety of products, has much contributed toward them reducing cost, improving reliability, enchancing performance, etc.

At the same time, with the expanding applications of a single-chip microcomputer, there have been increasing needs for higher speed, higher performance models.

The OLMS-64 Series, developed through CMOS silicon gate process technology, is a low power-consumption, high-speed and high-performance 4-bit single-chip microcomputer designed to meet these needs.

It can be used in areas that require high-speed processing for which conventional single-chip microcomputers have been inadequate and areas involving complex control. Its use in products is very effective in improving performance and reducing cost.

Areas of application

Portable video camera (VTR)
Audio equipment controller
Telecommunication equipment
Small-size printer, vending machine, etc.

1.2 FEATURES OF OLMS-64 SERIES

(1) 4k x 8 MASK ROM
The evaluation board is available for up to 8k x 8.
(2) 256 x 4 RAM (Including the stack area)
(3) 9 x 4 Ports 36 I/O lines
Four lines are input ports with a latch. The remaining 34 lines are all available for bit operation of I/O.
(4) Three built-in counters
12-bit time base counter
12-bit programmable timer
8-bit programmable timer/event counter
(5) Built-in 8-bit serial I/O register (with 3-bit counter)
(6) Five levels interrupts with priority (4 internal, 1 external)
(7) 32 stack registers (in RAM)
(8) Power down feature
(9) Instruction execution time 2μS @2MHz clock
(10) Set of instructions suitable for control
(11) Mask option
RC OSC/X'tal OSC
For port input
(12) Complete static operation
(13) Low poer consumption
(14) 5V single power supply, 42-pin DIP

1.3 DIFFERENCES BETWEEN OLMS-64 AND OLMS-40

The MSM6404, the highest version of the OLMS-64 Series, has higher performance than the MSM5840H, the highest version of the OLMS-40 Series, as shown below. MSM5840H features are shown in parentheses.

ROM/RAM	Double that of MSM5840H 4kx8/256x4 bits
I/O	36 lines (32 lines)
Execution time	About 1/4 times 2μS (7.6μS)
Built-in counters	3 times 3 built-in counters
8-bit serial I/O	Available (none)
Power down feature	Available (none)
Interrupt/stack	5 leveles/32 stacks (2 levels/4 stacks)

Also, the following software features are added to the MSM6404.

- Direct addressing of all RAM area
- Direct port addressing and indirect addressing by L register
- Bit operation/4-bit parallel comparison of all ports
- Masking for each interrupt
- Transmission of ROM table data to RAM
- Direct jump depending on memory and accumulator contents
- Vertical stacking instruction, rotate right instruction, flag register instruction
- Power down instruction
- Built-in counter control instruction

There is no instruction compatibility with the OLMS-40 Series, but most mnemonics are common and resemble closely.

1.4 PROGRAM DEVELOPMENT TOOL

The following are provided as OLMS-64 Series program development tools:

- OLMS-64 Series evaluation board
- CP/M-base cross assembler
- CP/M-base real time emulator
- General-purpose computer-base simulator

CHAPTER 2 OLMS-64 SERIES ARCHITECHTURE

2.1 GENERAL

The MSM6404, the highest version of the OMS-64 Series, integrates a 4-bit parallel CPU, a 4k x 8-bit mask ROM, a 256 x 4-bit RAM, 36 I/O lines, a 12-bit programmable timer, an 8-bit programmable timer/counter, a 12-bit time base counter, an 8-bit serial I/O register, five levels of priority interrupts, a clock oscillation circuit, etc.

The following pages explain the architecture of the MSM6404. When using another version, it is necessary to be careful about the differences in memory capacity, stack level, built-in I/O features, etc.

2.2 CPU ARCHITECTURE

Figures 1, 2 and 3 give a block diagram, pin arrangement and the logic symbols of the MSM6404, respectively.

What follows is an explanation of the architecture in Figure 1.

Like a conventional single-chip microcomputer, program and data address spaces are separated. That is to say, the ROM in which programs is addressed by the program counter, while the RAM in which data is addressed by the HL register.

The ROM consists of 4k x 8-bits. It can be addressed by the 12-bit program counter. It can also be addressed by a total of 12 bits of ACC and RAM contents by means of a special instruction.

The RAM consists of 16-column x 16-row x 4 bits. The 4 bits of the L register points a column address while the 4 bits of the H register points a row address.

Therefore, RAM data is selected by the HL register by addressing a row and column in the form of a matrix.

The MSM6404 instruction set permits efficient processing of multi-digits data arranged in two dimensions in the RAM.

This can be performed by using an instruction that makes automatic skipping when column or row addresses become "F" or "0", an instruction that makes an increment/decrement of the L register automatically at the time of the execution of the LOAD instruction, and a row address modification instruction.

The RAM can be addressed by the HL register. Besides, direct addressing is available by the second byte of an instruction. This permits using all the RAM areas as work or save registers.

The highest address (HL=FFH) toward a lower address of the RAM can also be used as a stack register area. The stack pointer SP points the first address of the stack register.

The arithmetic and logic circuitry (ALU), carry flag (C) and accumulator (ACC) are the core of data operation and transmission.

It is possible to perform 4-bit parallel operations and comparison, one-bit operation (set, reset, test) and rotation.

The F register consists of four flags which can be operated independently.

The decoder (DEC) decodes an instruction from the ROM and generates control signals necessary for its execution.

The DEC, ALU and SP cannot be operated by instructions.

For all ports, it is possible to perform 4-bit parallel inpu/output (only input for P2), comparison and single-bit processing. Also, P5 and 4 can provide 8-bit parallel output. Since all the 32 I/O lines are of the pseudo-bidirectional port structure, it is possible for each line to provide an input or output program.

P0 and P1 are for input/output of the built-in timer/counter and shift register.

With a one-byte input/output instruction, the port are addressed by the L register.

With a two-byte input/output instruction, it is possible to specify a port or bit position directly.

The internal control register controls the built-in timer/counter, shift register, interrupts, CPU start/stop, internal/external switching of the clock, masking of an interrupt cause, etc.

These registers are assigned to port addresses P9 to PD and can be operated like P0 to P8 by means of instructions.

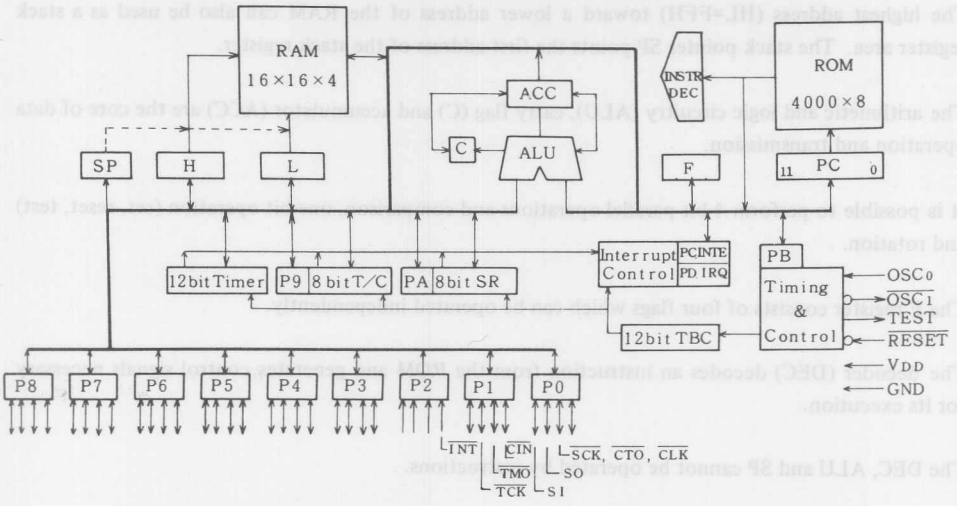


Fig. 1 MSM6404 Block Diagram

(Top View) 42 Lead Plastic DIP

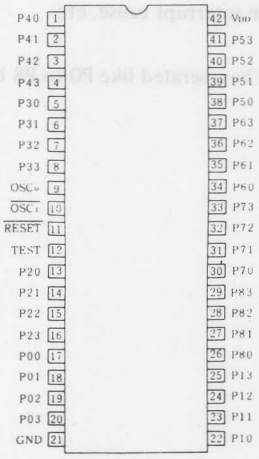


Fig. 2 Pin Arrangement

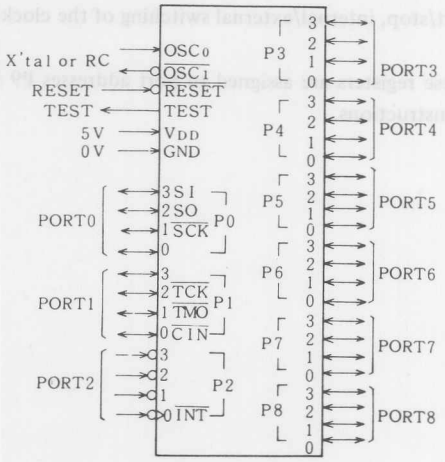


Fig. 3 Logic Symbols

2.2.1 Program Memory (ROM)

The program memory (hereinafter referred to as ROM) consists of 8 bits per word with a capacity of 4000 words. Figure 4 shows the structure of the ROM. A 64 word unit is called a page, and 4096 words are divided into 63 pages. 0 to 1FH of page 0 is the area that can be called by the zero page call instruction (CZP). Also, 40H, 42H, 44H, 48H and 50H are used as the first addresses of the respective interrupt routines. If these are not used for this purpose, these can be used as ordinary program areas.

Addresses FA0H (4000) through FFFH (4095) are used for test purposes and cannot be used as a program area.

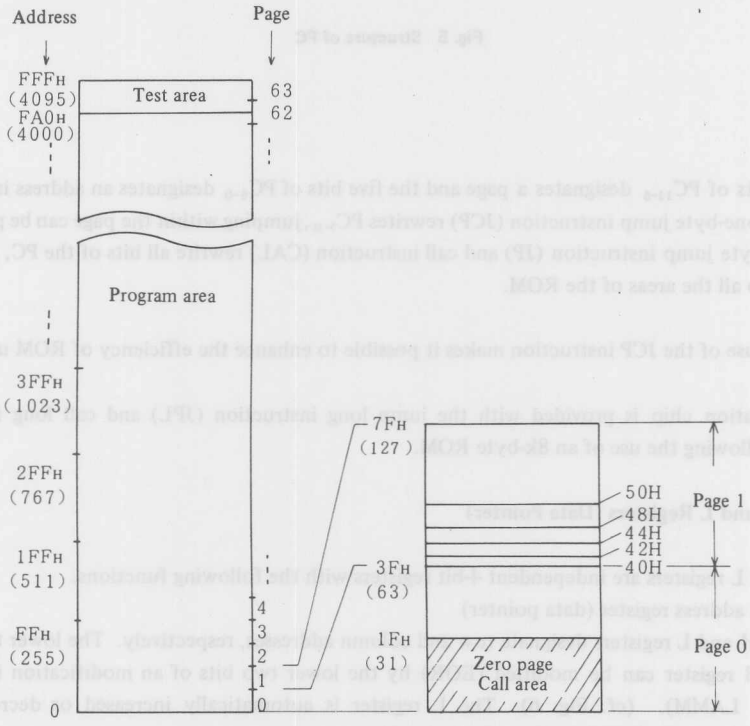


Fig. 4 ROM Structure

Address 0 of the ROM is used by the CZP instruction (CZP 0) and also used as the program start address at the time of resetting.

2.2.2 Program Counter (PC)

The program counter (hereinafter referred to as PC) shows a ROM address. Everytime an instruction has been executed, the PC shows the address of the next instruction. Figure 5 shows the structure of the PC.

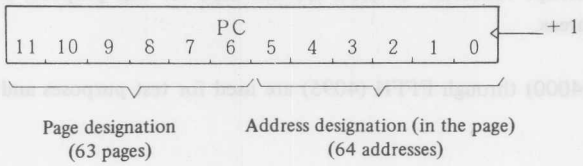


Fig. 5 Structure of PC

The six bits of PC_{11-6} designates a page and the five bits of PC_{5-0} designates an address in the page. Since the one-byte jump instruction (JCP) rewrites PC_{5-0}, jumping within the page can be performed. The two-byte jump instruction (JP) and call instruction (CAL) rewrite all bits of the PC, permitting jumping to all the areas of the ROM.

Frequent use of the JCP instruction makes it possible to enhance the efficiency of ROM use.

The evaluation chip is provided with the jump long instruction (JPL) and call long instruction (CALL), allowing the use of an 8k-byte ROM.

2.2.3 H and L Registers (Data Pointer)

The H and L registers are independent 4-bit registers with the following functions.

(a) RAM address register (data pointer)
The H and L registers designate row and column addresses, respectively. The lower two bits of the H register can be modified (EOR) by the lower two bits of an modification instruction (XM, LAMM). (cf. Fig. 6) The L register is automatically increased or decreased with execution of exchange and transfer instruction.

(b) Port address register
The L register designates the port address at the time of the execution of an input/output instruction (one-byte instruction).

(c) General purpose register
The H and L registers each can be used as a general purpose register except when designating RAM and port addresses.

The following are the H/L register-related instructions.

- Immediate load
- Data transmission with ACC
- Increment (with skip)
- Decrement (with skip)
- Comparison with L register and immediate value (with skip)

The content of the H/L register becomes "0" at the time of resetting.

2.2.4 Data Memory (RAM)

The data memory is a 4k-bit static RAM. Its word arrangement is made in the form of 16 columns x 16 rows with 4 bits being one word. Figure 6 shows the structure of the RAM. Row-direction addresses (sometimes called a file) are designated by the H register, while column-direction addresses are designated by the L register. It is also possible to designate an address directly by the second byte of an instruction.

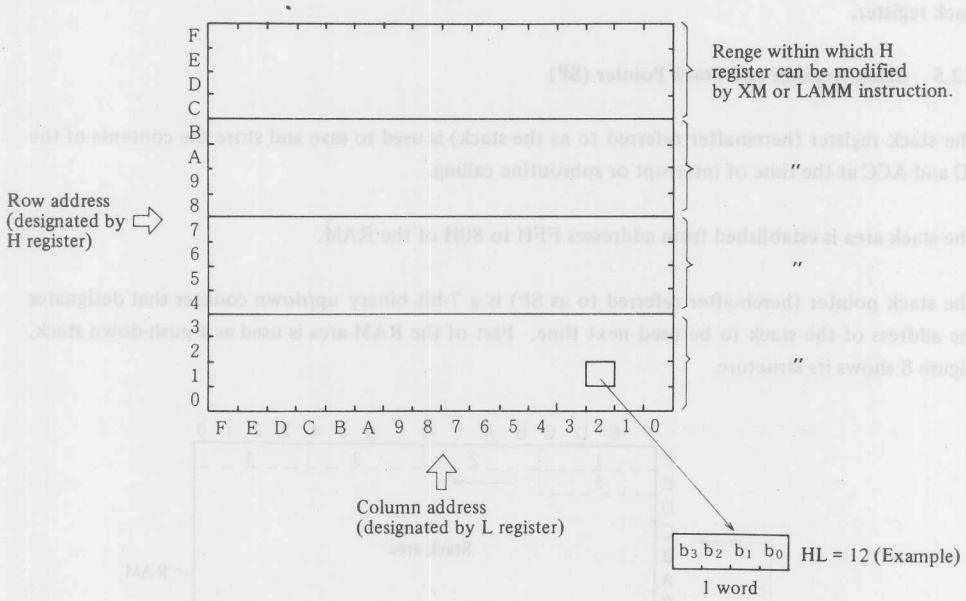


Fig. 6 Structure of RAM

The following operations are available for one word (4 bits) selected by the H/L register.

- Data transmission with ACC
- Increment (with skip)
- Decrement (with skip)
- Bit operation (set, reset, test)

Also, the following operations are available with an instruction that designates a RAM address directly by the secod byte of the instruction. (The H/L register is not destructed.)

Data transmission with ACC
Increment (with skip)
Decrement (with skip)

With an instruction that handles two words (8 bits) in a batch, the following operations can be performed.

Immediate load
Data transmission with the built-in counter, timer and shift register
Table data (ROM data) transmission

The 8 bits can be handled through addressing with the lowest bit of the L register ignored. For details, refer to the rules of instructions (P49).

The RAM addresses from the highest address (HL=FFH) downward to 80H can also be used as a stack register.

2.2.5 Stack Register and Stack Pointer (SP)

The stack register (hereinafter referred to as the stack) is used to save and store the contents of the PC and ACC at the time of interrupt or subroutine calling.

The stack area is established from addresses FFH to 80H of the RAM.

The stack pointer (hereinafter referred to as SP) is a 7-bit binary up/down counter that designates the address of the stack to be used next time. Part of the RAM area is used as a push-down stack. Figure 8 shows its structure.

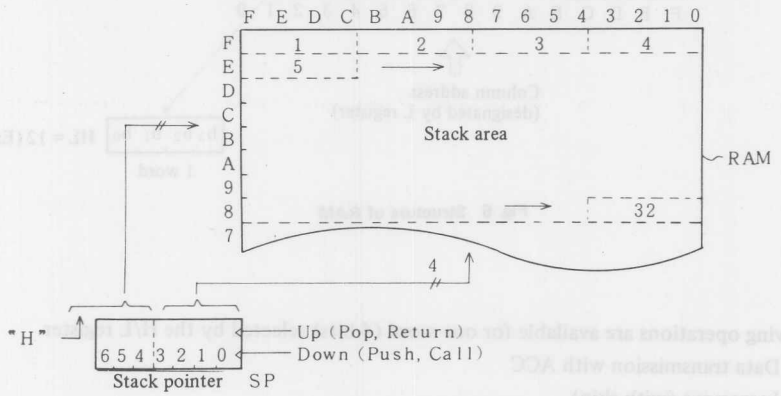


Fig. 8 Structure of Stack

The SP is set so that it designates address FFH of the RAM at the time of resetting. It counts down (–4) with the execution of a call instruction (CZP, CAL or CALL) and a push instruction, and counts up (+4) with the execution of a return instruction (RT or RTS) and a pop instruction.

That is, when an interrupt or call instruction is executed, the content of the PC at that time is saved to the four words (16 bits) of the RAM, and the saved content is returned to the PC with the execution of a return instruction. If a push instruction is executed, the contents of the ACC, C, H and L are saved to the stack. The execution of a pop instruction causes the saved contents to be returned to the respective registers. Figure 9 gives an example.

Since the stack area has 128 words (FFH to 80H), up to 32 levels of stack nesting are available. Nesting of more than 32 levels causes the SP content to be FFH, destructing the content of the first stack register.

If there spare stack levels, the area can be used as a RAM. For example, if stack levels are limited to 4 levels, FFH to FOH become the stack area and the rest can be used as an ordinary RAM.

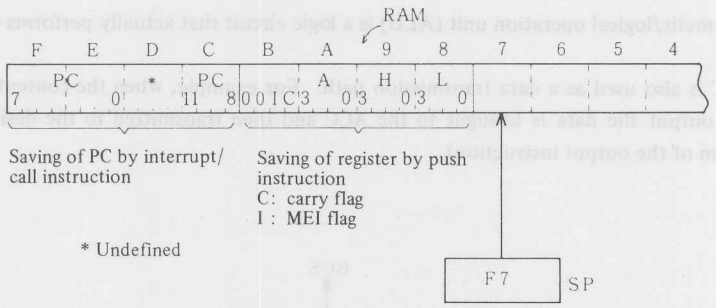


Fig. 9 Example of Saving within Stack

2.2.6 F Register (FR)

The F register (FR) consists of four independent flip-flops. Each flip-flop can perform setting, resetting and testing by means of a one-byte instruction. However, it is impossible to perform four-bit parallel data transmission.

The FR can be used freely as a "flag" on the program. At the time of SYSTEM resetting, the content of each flag becomes "0".

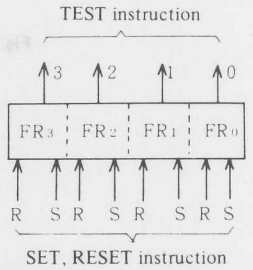


Fig. 10 Structure of F Register

2.2.7 Arithmetic Unit (ACC, C and ALU)

The arithmetic unit performs operations between the content of the meory pointed by the H/L register and the content of the accumulator (ACC). The result is obtained in the ACC. There are the following types of operations:

- 4-bit parallel artithmetic operations
  (addition and subtraction . . . with skip)
- 4-bit parallel logical operations
  (AND, OR, and EOR)
- 4-bit comparison (with skip)
- Decimal correction
- Right/left rotation (one bit)
- Increment/decrement (with skip)

In an arithmetic operation, if a carry occurs, carry flag C is set. (In the case of an operation with a carry).

The arithmetic/logical operation unit (ALU) is a logic circuit that actually performs operations.

The ACC is also used as a data transmission path. For example, when the content of the memory is to be output the data is brought to the ACC and then transmitted to the desired output port (execution of the output instruction).

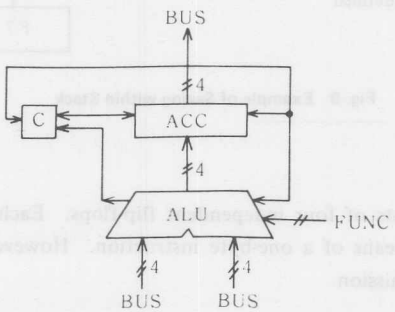


Fig. 11 Structure of Arithmetic Unit

2.2.8 Stop/Halt Mode

Setting bit b_1 of port PB, that is, executing a STOP instruction, causes a stop state. In this mode, the clock stops oscillating, all the operations of the CPU stops and power consumption is minimized. However, the 12-bit timer and shift register is operational, provided the external clock is selected.

A stop state is cleared by an external interrupt input of P20, timer interrupt or reset input. At this time, an input width which is more than the time required for stabilized oscillation is necessary. This signal must not has chattering.

Figure 12 shows this timing.

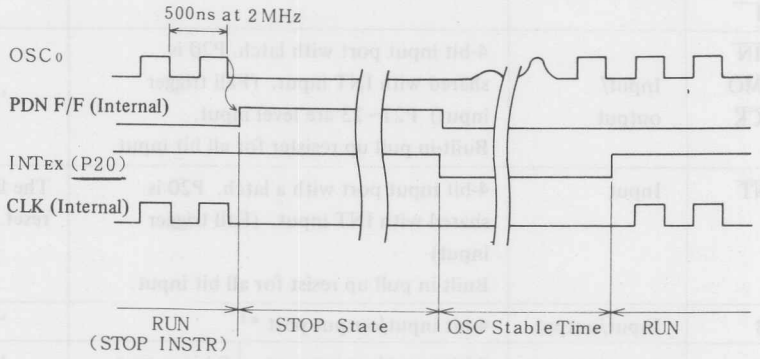


Fig. 12 Timing of Stop Mode Clearing

The figure shows an example of resetting a stop mode by an external interrupt. When resetting a stop mode by an internal timer, it is necessary to determine the timer clock (external) cycle by taking the time required for stabilized oscillation into consideration. That is, it is necessary to make the timer clock (external) cycle longer than the time required for stabilized oscillation.

When clearing a stop state by an interrupt, if an MDI state exists, the next instruction is executed. If an MEI state exists, an interrupt is accepted. (See P86 STOP instruction.)

Setting bit b_0 of port PB causes a halt mode. In this state, the clock does not stop oscillating and only the execution of an instruction halts.

In a halt state, no program counter increment is made and therefore, the NOP instruction is executed repeatedly at the address next to the HALT instruction. Power consumption decreases to some extent.

A halt mode is cleared by all interrupts in an EI state or by reset input. In the case of reset input, execution restarts from address 0. In the case of interrupt, execution restarts from the address of each interrupt. If a halt state occurs in an MDI state, execution restarts from the next instruction of HALT with interrupt request on EIF=1. (See P85 HALT instruction.)

2.3 TERMINAL FUNCTIONS

Table 1 gives a list of the MSM6404RS terminal functions.

Table 1 List of Terminal Functions

Terminal name	Input/ output	Function	When reset
P00 P01 /$\overline{SCK}$ P02 /$\overline{SO}$ P03 /$\overline{SI}$	Input/ output	4-bit input/output port. P01 to P03 are also used "1" as serial interface terminals.	"1"
P10 /$\overline{CIN}$ P11 /$\overline{TMO}$ P12 /$\overline{TCK}$ P13	Input/ output	4-bit input port with latch. P20 is shared with INT input. (Fall trigger input) P21~23 are level input. Built-in pull up resister for all bit input.	"1"
P20 /$\overline{INT}$ P21 P22 P23	Input	4-bit input port with a latch. P20 is shared with INT input. (Fall trigger input) Built-in pull up resist for all bit input.	The latch is reset.
P30–33	Input/output	4-bit input/output port *1.	"1"
P40–43	Input/output	4-bit input/output port 8-bit output	"0"
P50–53	Input/output	4-bit input/output port port	"0"
P60–63	Input/output	4-bit input/output port*1	"0"
P70–73	Input/output	4-bit input/output port	"0"
P80–83	Input/output	4-bit input/output port	"0"
OSC_0 $\overline{OSC_1}$	Input/output	X'tal/RC connection terminal for system clock oscillation *2	Oscillation wave
TEST	Output	(Test terminal for Maker)	Pulse output
$\overline{RESET}$	Input	System reset input terminal	
VDD GND		Power source voltage supply	

Note: When each port is used for output, it is possible to drive one LSTTL (one input).

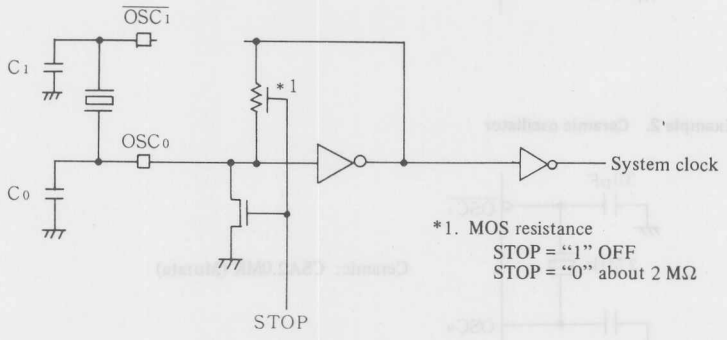
*1. Can be made as a port dedicated to input (mask option).

*2. RC oscillation is a mask option.

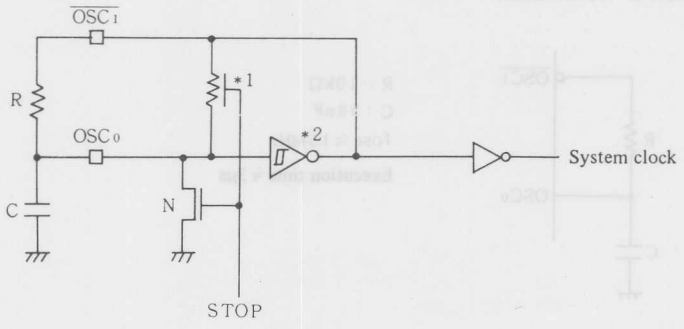
2.3.1 Clock Oscillation Terminal (OSC_0, $\overline{OSC_1}$)

A built-in inverter is provided for system clock oscillation.

The system clock is oscillated by connecting a quartz oscillator or a ceramic resonator between the OSC_0 and $\overline{OSC_1}$ terminals. Figure 13 shows the structure of the system clock. When the system clock stops, the feedback resistance of the inverter is made OFF and input is made "0" to prevent power consumption.



(a) In the case of quartz or ceramic resonator

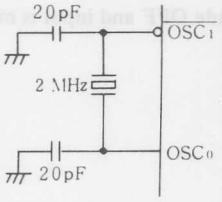


(b) In the case of RC element (mask option)

Fig. 13 Generation of System Clock

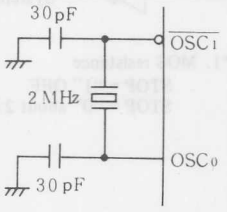
In the case of RC oscillation, it is possible to stop the clock, but there is some current through the external resistance R. Figure 14 gives examples of system clock oscillation.

Example 1. Qualtz oscillator



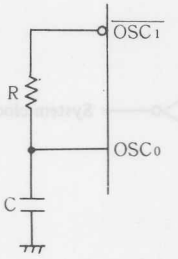
X'tal : KSS HC 18/U

Example 2. Ceramic oscillator



Ceramic: CSA2.0MK (Murata)

Example 3. RC oscillation



R : 10kΩ
C : 30pF
$f_{OSC} \fallingdotseq 1.5$MHz
Execution time ≒ 3μs

Fig. 14 Examples of System Clock Oscillation

Figure 15 shows the relationship between the RC constant and frequency in the case of RC oscillation.

With RC oscillation, the frequency varies with internal properties of the IC and individual RCs. It is necessary to set the constant so that it does not exceed the specified value even in the worst case.

In the cases of a qualtz and a ceramic oscillator, too, it is necessary to set the constant after determining the oscillation properties.

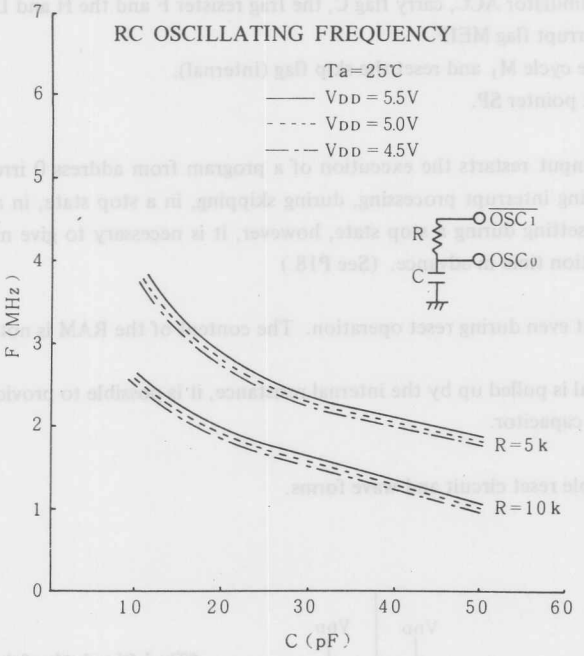


Fig. 15 RC Oscillating Frequency (Typical value)

2.3.2 Reset Terminal ($\overline{\text{RESET}}$)

This is the input terminal that reset the inside of the IC. When the system clock is supplied, the terminal becomes effective by giving two or more machine cycles of the "0" level to it.

Reset input overrides any other signal, providing the following processing:

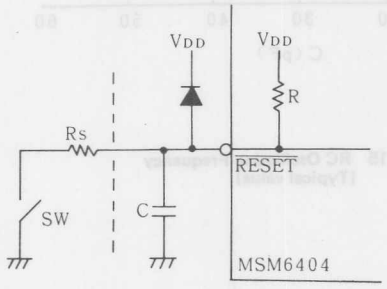
Make all the bits of the program counter "0".
Reset internal ports P9 to PD.
Set ports P0, P1 and P3. (Output "1".)
Reset prots P4, 5, 6, 7, and 8. (Output "0".)
Reset the latch of port P2.
Reset (clear) accumulator ACC, carry flag C, the frag resister F and the H and L registers.
Reset master interrupt flag MEIF.
Make the machine cycle M_1 and reset the skip flag (internal).
Set "FF" in stack pointer SP.

That is to say, reset input restarts the execution of a program from address 0 irrespective of the current state, e.g., during interrupt processing, during skipping, in a stop state, in a halt state, and so on. To perform resetting during a stop state, however, it is necessary to give more reset input time than clock oscillation time in advance. (See P18.)

A TEST signal is output even during reset operation. The content of the RAM is not affected.

Since the reset terminal is pulled up by the internal resistance, it is possible to provide auto reset by connecting an external capacitor.

Figure 16 shows a sample reset circuit and wave forms.



*The lefthand side of the broken line is a manual reset connection example.

Fig. 16 (a) Reset Circuit

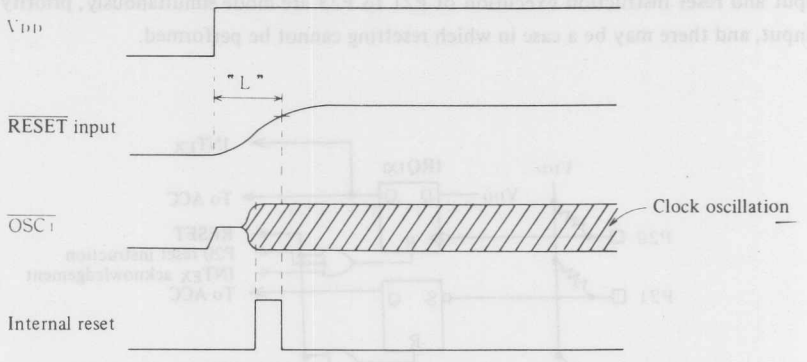


Fig. 16 (b) Wave Forms

As shown in the figure, after power is supplied, the internal reset becomes effective when two machine cycles of the clock have been supplied after the start of the oscillation of the oscillating circuit. The reset is released when the RESET input reaches the "H" level.

Therefore, like the resetting during a stop state described previously, therefore, the rise wave form of the RESET input requires keeping the "L" level for more than the oscillating time (rigorously plus two machine cycles) of the oscillation circuit. Also, after power supply until the reset becomes effective, the port cannot be initialized and for this reason, output is unknown "H" or "L".

2.3.3 Input/Output Ports (P0 to P8)

Since port addresses are specified by the L register, there are logically 16 ports (0 to F). Of these, port addresses 0 to 8 are assigned as input/output ports and 9 to D are assigned as internal ports (counter, timer and interrupt control registers). Port addresses E and F are vacant codes and no corresponding hardware exists.

There are a total of 36 input/output ports from P00 to P83. (See P19, Table 1.) P0 and P1 Terminals are used for shift register and counter input/output. These Terminals are selected by the internal port.

P2 is an input port with a latch. P20 is set by a falling of edge of input, and P21 to 23 are set by level input.

Figure 17 shows the structure of P2. P20 is shared with the interrupt request flag (IRQ_{EX}). If this port is set and an interrupt occurs, resetting is made automatically.

If input and reset instruction execution of P21 to P23 are mode simultanously, priority is given to the input, and there may be a case in which resetting cannot be performed.

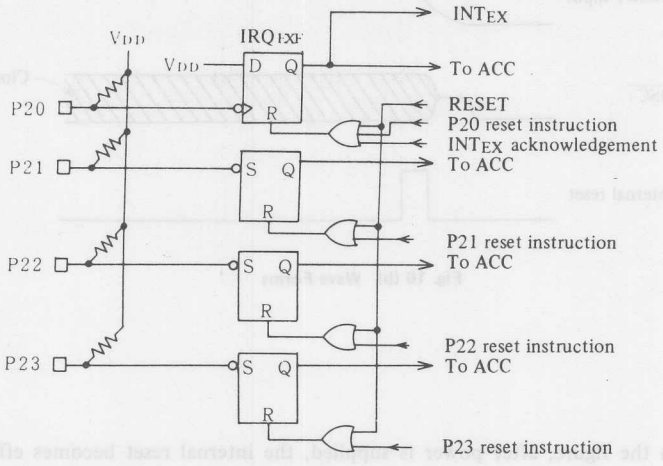


Fig. 17 Structure of P2

P4 and P5 are also used as a pair for output of 8-bit data (ROM data). A 7-segment indicator is convenient if connected to this port.

All the ports except P2 are pseudo bidirectional ports.

Figure 18 shows the internal structure of the input/output port.

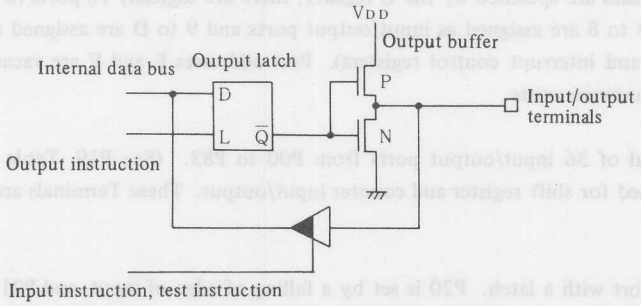
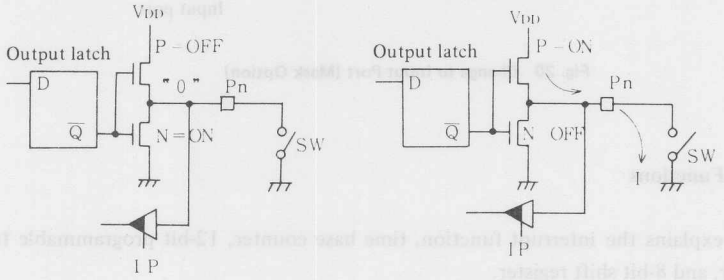


Fig. 18 Structure of Input/Output Port

If this port is used as an input terminal, it is necessary to write "1'. in the output latch to set the output buffer N-channel transistor off. In this case, the P-channel transister is set on and pulled up with the ON resistance (about 10K). Usually, therefore, it is unnecessary to connect an external pull-up resistance.

The port can also be used in the following way. . That is, when inputting switch and sensor information, write "0" in the output latch and input the information by writing "1" only when necessary, and write "0" after the input and thereby minimize the power consumption by the pull-up resistance.



(1) Normally, "0" is output and the Pn line is kept at the GND level.

(2) "1" is output only when switch data is read. If the switch is off, the "1" level is read. Otherwise, the "0" level is read. At this time the following current is applied.

$$I = \frac{V_{DD}}{R_{ON}} \doteqdot \frac{5V}{10k\Omega} = 0.5\ mA$$

(3) "0" is output again and the Pn line is made the GND level.

Suppose the time of the state (2) is 10 μs and the switch data input cycle is 10 ms, the average current I_{AV} applied is as follows:

$$I_{AV} \doteqdot 0.5\ mA \times \frac{10\mu S}{10mS} \doteqdot 0.5\mu A$$

As described above, each pseudo bidirectional port can be programmed as an input, output, or input/output terminal and can be interfaced with an external device with a low power consumption.

At the time of resetting, P0, P1 and P3 are initialized in an input state (output latch = "1").

There is some variance of the ON resistance of the P channel FET. Therefore, to design an external circuit requires checking the electrical properties.

If a pseudo bidirectional port is inconvenient for interface purposes, it is possible to use only P3 and P6 as dedicated input ports by separating both P and N FETs through a mask change.

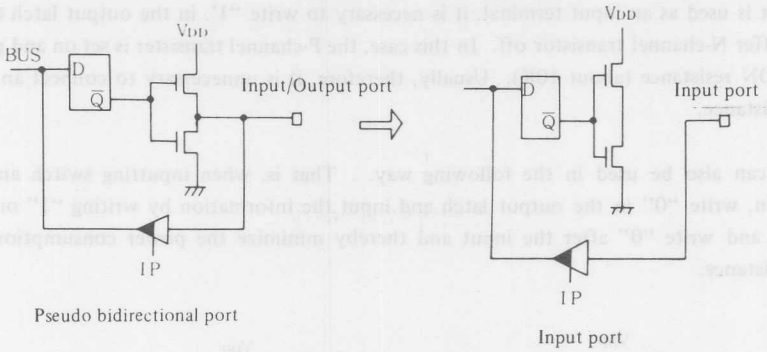


Fig. 20 Change to Input Port (Mask Option)

2.4 Built-in Functions

This section explains the interrupt function, time base counter, 12-bit programmable timer, 8-bit timer/counter, and 8-bit shift register.

2.4.1 Interrupt Function

With the MSM6404, interrupts are available in five levels in five types. Internal ports PC and PD are provided as registers to control these interrupts. PC controls the inhibition and permit of each interrupt and is called the enable interrupt flag (EIF). PD is a register that retains each interrupt request until it is accepted and is called the interrupt request flag. (IRQF) The IRQF of external interrupt is shared with P20. Also, the master enable interrupt flag (MEI) is provided to control the entire interrupt inhibition/permit. Figure 21 shows a block diagram of the interrupt control circuit. Table 2 shows the IRQF, EIF, interrupt address and priority order for each interrupt request.

The operation of IRQF and IEF is the same as that of ports. The memonic corresponding to each is provided. (See P43.)

Table 2 Interrupt Requests and Corresponding Flags

Interrupt		Priority order	IRQ	EIF	Interrupt address
External interrupt	(EXINT)	1	P20 (IRQ EX)	PC0 (EIEX)	50H
Time base interrupt	(TBINT)	2	PD0 (IRQ TB)	PC1 (EITB)	48H
Timer interrupt	(TMINT)	3	PD1 (IRQ TM)	PC2 (EITM)	44H
Counter interrupt	(CTINT)	4	PD2 (IRQ CT)	PC3 (EICT)	42H
Shift magnet interrupt	(SRINT)	5	PD3 (IRQ ST)	PC3 (EICT)	40H

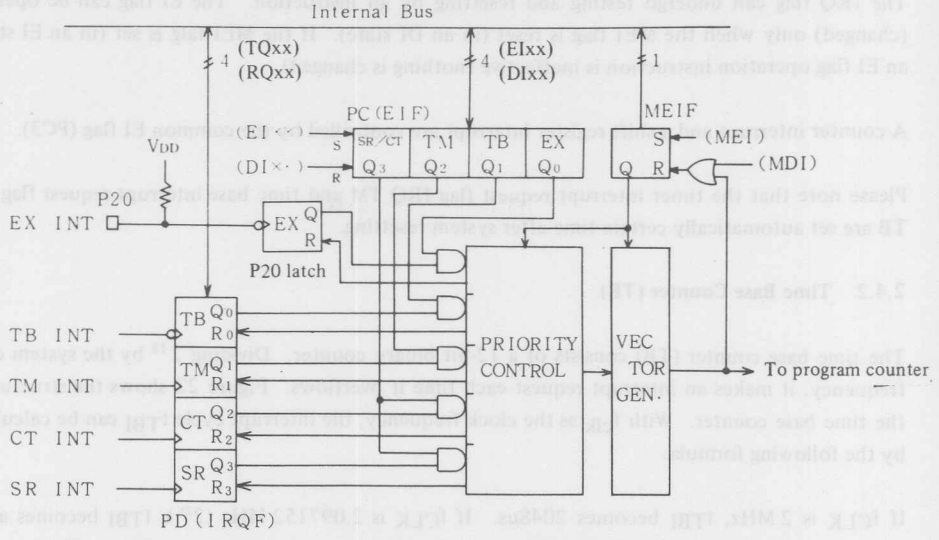


Fig. 21

These interrupts occur in the following cases:

External interrupt	P20 pulse falling edge
Time base interrupt	Time base counter overflow
Timer interrupt	Coincidence signal of timer content and preset value (12 bits)
Counter interrupt	Coincidence signal of count value and preset value (8 bits)
Shift register interrupt	End signal of 8-bit shift

With these signals, the corresponding IRQ flags are set. At this time, if the corresponding EI flag is set and the MEI flag is also set, that is, if an interrupt is available, interrupt operations are preformed in the following sequence:

- ○
- ○ Reset the MEI flag and prohibit all the subsequent interrupt.
- ○ Save the content of the program counter and subtract 4 from the content of the stack pointer.
- ○ Load the address corresponding to the interrupt request in the program counter. At the same time, reset the IRQ flag that has made the interrupt request.

The time required for the above is three machine cycles (6 μs@2 MHz).

If several interrupt requests occur simultaneously, the request with the highest priority is accepted first and the rest are held in the IRQF.

The IRQ flag can undergo testing and resetting by an instruction. The EI flag can be operated (changed) only when the MEI flag is reset (in an DI state). If the MEI falg is set (in an EI state), an EI flag operation instruction is ineffective (nothing is changed).

A counter interrupt and a shift register interrupt are controlled by the common EI flag (PC3).

Please note that the timer interrupt request flag IRQ TM and time base interrupt request flag IRQ TB are set automatically certain time after system resetting.

2.4.2 Time Base Counter (TB)

The time base counter (TB) consists of a 12-bit binary counter. Dividing 2^{12} by the system clock frequency, it makes an interrupt request each time it overflows. Figure 22 shows the structure of the time base counter. With f_{clk} as the clock frequency, the interrupt cycle t_{TBI} can be calculated by the following formula:

If f_{CLK} is 2 MHz, t_{TBI} becomes 2048μs. If f_{CLK} is 2.097152 MHz (2^{21}), t_{TBI} becomes about 1953.2 μs. If the former is 2.048 MHz, then the latter becomes just 2 ms. Figure 23 shows the TBC timing.

The time base interrupt cycle can be used as the time base of key input or display scanning and the clock.

The middle stage output of the time base counter is supplied as the clock pulse for the internal timer, counter and shift register.

The time base counter is always active and cannot be stopped. Therefore, the time base interrupt request IRQ TB flag is set certain time after resetting. However, if the clock is stopped by a stop instruction, the time base counter also stops.

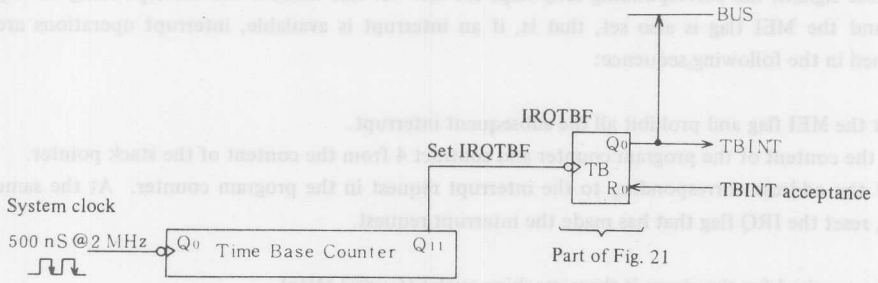


Fig. 22 Structure of Time Base Counter

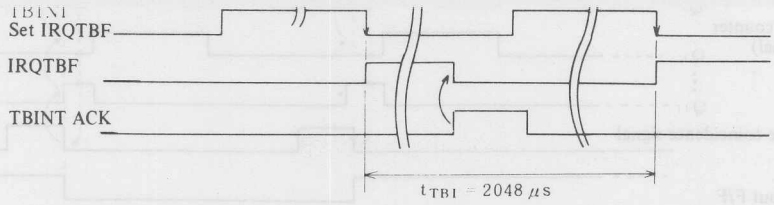


Fig. 23 TBC Timing

2.4.3 Programmable Timer (TM)

The programmable timer (TM) consists of a 12-bit register, binary up-counter, comparison circuit and control circuit. Figure 24 shows its construction.

The internal port P90 permits selecting the clock pulse "external/internal", while P93 permits selecting the cycle 128 μ/8 μs. If the P90 selects "external" (P90=1), then the input pulse to P12 becomes the timer clock. At this time, it is necessary to write "1" in the output latch of P12. It is also possible to make a timer clock by writing "1" and "0" in the output latch of P12 alternately.

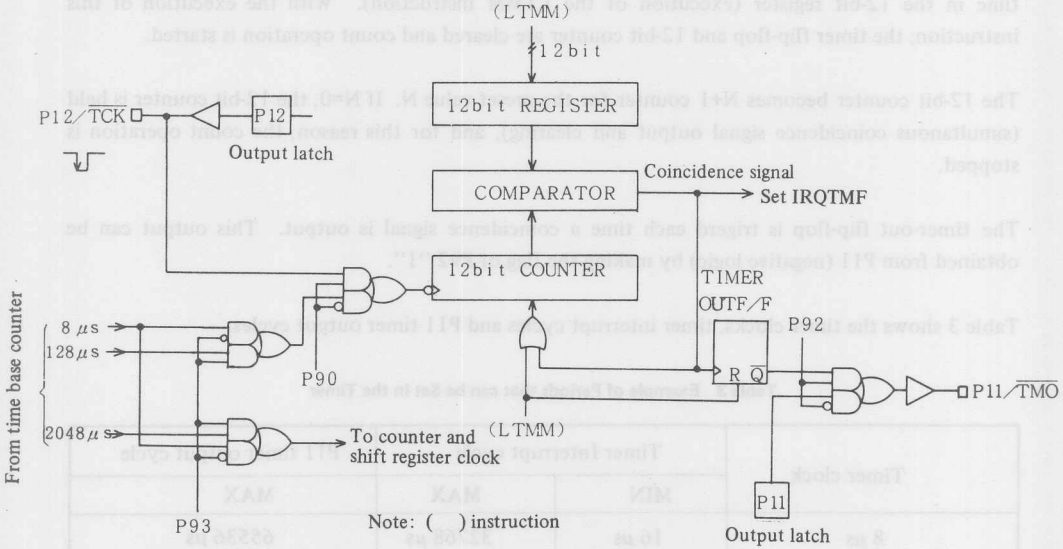


Fig. 24 Construction of Programmable Timer

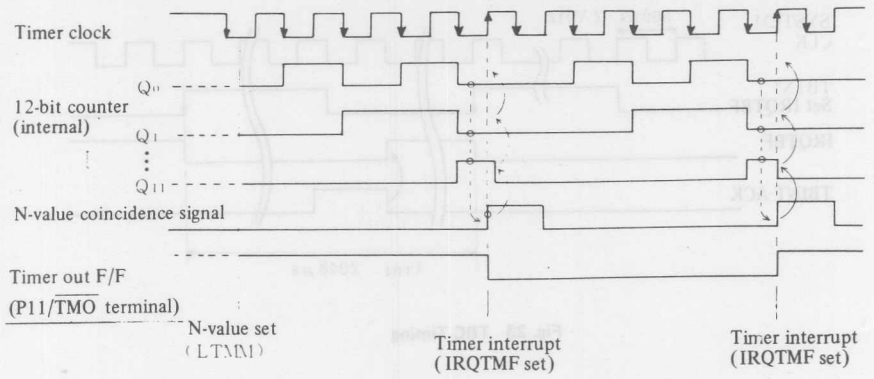


Fig. 25 Example of Timer Timing

The 12-bit counter has an increment each time the timer clock pulse falls. If the preset value N to the 12-bit register and the counter content become identical, a coincidence signal is made in synchronization with the rise of the next clock pulse, trigering the timer-out flip-flop, and a timer interrupt occurs. At the same time, the 12-bit counter is reset and the count operation is repeated. Figure 25 shows a time chart of the timer.

The timer is started by setting the value N (N=001H to FFFH) that is used to obtain the desired time in the 12-bit register (execution of the LTMM instruction). With the execution of this instruction, the timer flip-flop and 12-bit counter are cleared and count operation is started.

The 12-bit counter becomes N+1 counter for the preset value N. If N=0, the 12-bit counter is held (simultanous coincidence signal output and clearing), and for this reason, the count operation is stopped.

The timer-out flip-flop is trigerd each time a coincidence signal is output. This output can be obtained from P11 (negative logic) by making the flag of P92 "1".

Table 3 shows the timer clocks, timer interrupt cycles and P11 timer output cycles.

Table 3 Example of Periods that can be Set in the Timer

Timer clock	Timer Interrupt cycle		P11 timer output cycle
	MIN	MAX	MAX
8 μs	16 μs	32768 μs	65536 μs
128 μs	256 μs	524288 μs	1048576 μs
2048 μs*	4096 μs	8388608 μs	16777216 μs

* Example for external input to P12/$\overline{\text{TCK}}$ terminal

2.4.4 Timer/Event/Counter (CT)

The 8-bit timer/event counter (CT) consists of an 8-bit register, 8-bit binary up-counter, comparison circuit and control circuit. Figure 26 shows the construction of the 8-bit timer counter.

The counter clock can select "internal/external" with P91, and the cycle (2048μs/8μs) with P93. If P91 selects "external" (P91="1"), the input pulse to P10 becomes the counter clock. At this time, it is necessary to write "1" in the output latch of P10. It is also possible to make a counter clock by writing "1" and "0" in the output latch of P10 alternately.

The 8-bit counter has an increment with the fall of the internal clock and the external clock. However, the counter stops with the execution of STOP.

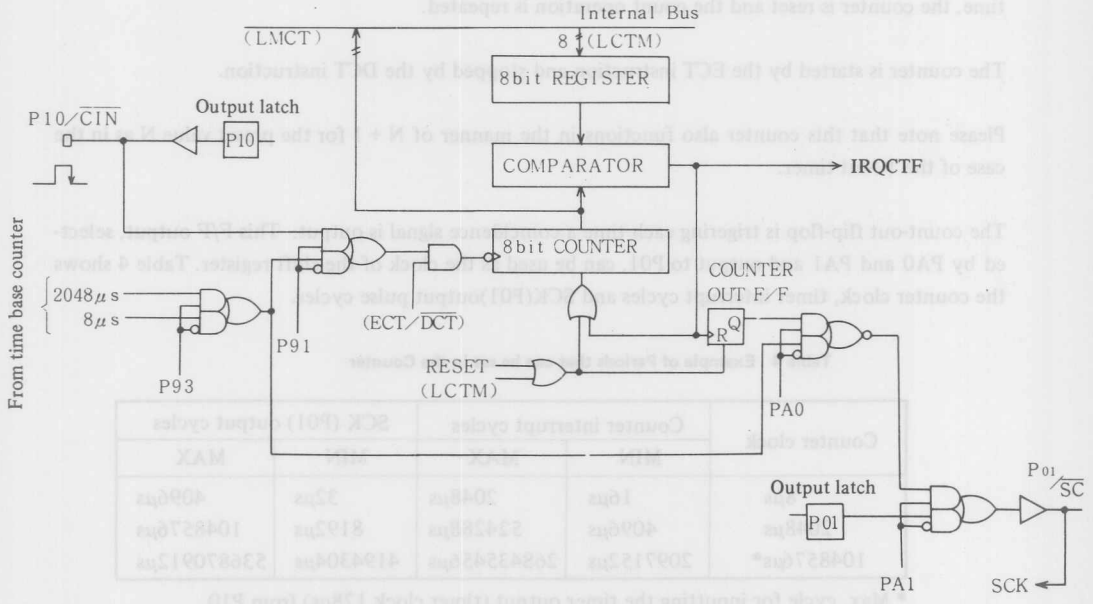


Fig. 26 Construction of 8-bit Timer/Counter

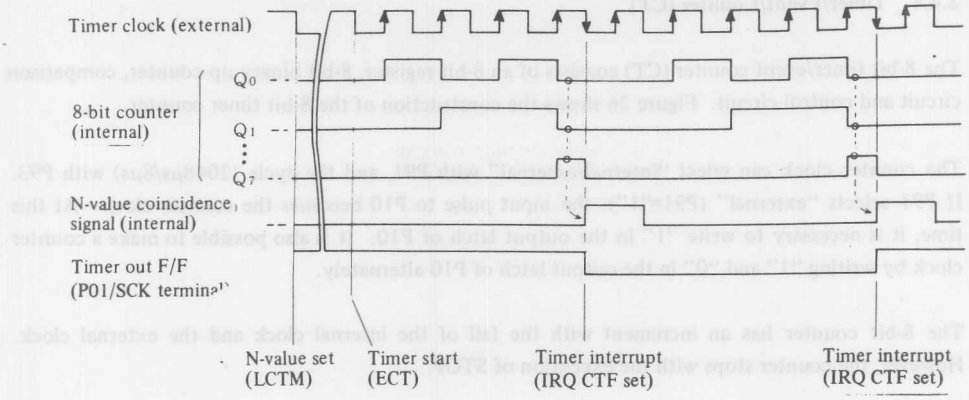


Fig. 27 Example of Counter Timing

Figure 27 shows an example of counter timing.

If the preset value N to the 8-bit register becomes equal to the content of the counter, a coincidence signal is made, trigering the counter-out flip-flop and setting the counter interrupt flag. At the same time, the counter is reset and the count operation is repeated.

The counter is started by the ECT instruction and stopped by the DCT instruction.

Please note that this counter also functions in the manner of N + 1 for the preset value N as in the case of the 12-bit timer.

The count-out flip-flop is trigering each time a coincidence signal is output. This F/F output, selected by PA0 and PA1 and output to P01, can be used as the clock of the shift register. Table 4 shows the counter clock, timer interrupt cycles and SCK(P01)output pulse cycles.

Table 4 Example of Periods that can be set in the Counter

Counter clock	Counter interrupt cycles		SCK (P01) output cycles	
	MIN	MAX	MIN	MAX
8μs	16μs	2048μs	32μs	4096μs
2048μs	4096μs	524288μs	8192μs	1048576μs
1048576μs*	2097152μs	268435456μs	4194304μs	536870912μs

* Max. cycle for inputting the timer output (timer clock 128μs) from P10.

2.4.5 Shift Register (SR)

The shift register (SR) consists of an 8-bit shift register, 3-bit shift counter and control circuit. It is used for serial data input/output. Figure 28 shows its construction.

Serial data input/output is performed in synchronization with the shift clock. The shift clock pulse is given to the P01/$\overline{SCK}$ terminal. At this time, it is necessary to write "1" in the output latch of P01. If the internal clock is used, it is possible to select 8μs, 2048μs, or any cycle programmed in the counter (CT) by PA0 and PA1. It is also possible to make a shift clock by writing "1" and "0" alternately in the output latch of P01.

Figure 29 shows the timing of serial transmission.

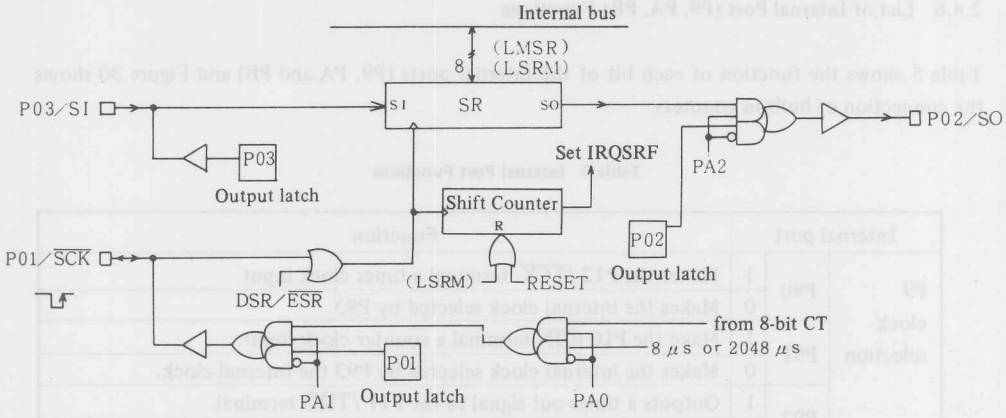


Fig. 28 Construction of Shift Register

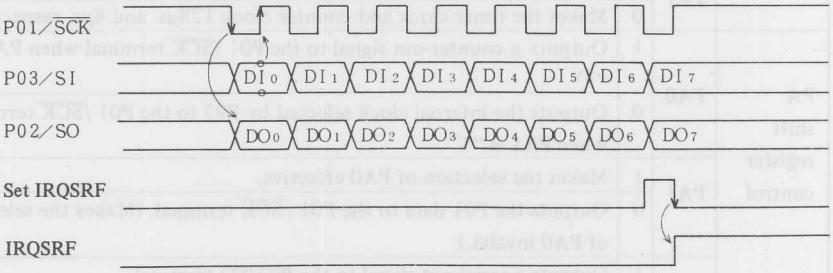


Fig. 29 Timing of Serial Transmission

With the fall of the shift clock (SCK), the lowest bit (bit 0) of the shift register is output to the P02/SO terminal. With the rise of the shift clock, the content of the shift register is shifted by one bit to the right (bit n → bit n-1). At the same time, the data of the P03/SI is loaded to the highest bit of the shift register.

The shift counter counts the shift clock. If it counts 8, it sets the shift register interrupt request flag IRQSRF and at the same time resets the shift register control flag SRF. Therefore, even if the shift clock is applied continuously, only the first 8 pulses are effective.

The shift register is started by the ESR instruction. It is stopped automatically as described above, but it can also be stopped by the DSR instruction.

It is possible to preset 8-bit data in the shift register only when it is in a stop state. If it is started, it is only possible to read data.

When using an external shift clock, it is necessary to give the shift clock after starting the shift register. If the shift clock is started while giving the external shift clock, the shift operation may become inaccurate because they are synchronous.

2.4.6 List of Internal Port (P9, PA, PB) Functions

Table 5 shows the function of each bit of the internal ports (P9, PA and PB) and Figure 30 shows the connection of built-in counters.

Table 5 Internal Port Functions

Internal port			Function
P9 clock selection	P90	1	Makes the P12 /$\overline{\text{TCK}}$ terminal a timer clock input.
		0	Makes the internal clock selected by P93.
	P91	1	Make the P10 /$\overline{\text{CIN}}$ terminal a counter clock input.
		0	Makes the internal clock selected by P93 the internal clock.
	P92	1	Outputs a timer-out signal to the P11 /$\overline{\text{TMO}}$ terminal.
		0	Outputs the P11 data to the P11 /$\overline{\text{TMO}}$ terminal.
	P93	1	Makes the timer clock 128μs, and makes the counter clock 2048μs.
		0	Makes the timer clock and counter clock 128μs and 8μs, respectively.
PA shift register control	PA0	1	Outputs a counter-out signal to the P01 /$\overline{\text{SCK}}$ terminal when PA1 is "1".
		0	Outputs the internal clock selected by P93 to the P01 /$\overline{\text{SCK}}$ terminal when PA1 is "1".
	PA1	1	Makes the selection of PA0 effective.
		0	Outputs the P01 data to the P01 /$\overline{\text{SCK}}$ terminal. (Makes the selection of PA0 invalid.)
	PA2	1	Outputs a serial-out signal to the P02/SO terminal.
		0	Outputs the P02 data to the P02/SO terminal.
	PA3	1	Not used
		0	
PB CPU control	PB0	1	Halts the CPU. (Clearing is made by an interrupt or reset input.)
		0	Normal operation
	PB1	1	Stops the CPU (stopping OSC). Clearing is made by an external or timer interrupt or reset input.
		0	Normal operation
	PB2	1	Shift register start ESR
		0	Shift register stop DSR
	PB3	1	Counter start ECT
		0	Counter stop DCT

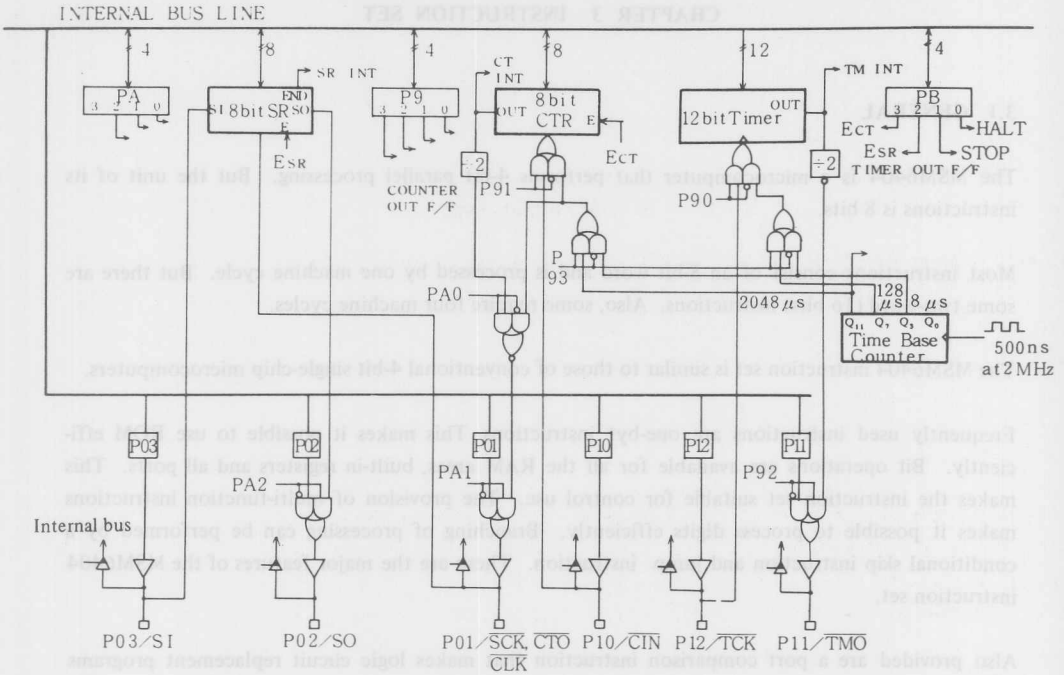


Fig. 30 Built-in Counter Connection

CHAPTER 3 INSTRUCTION SET

3.1 GENERAL

The MSM6404 is a microcomputer that performs 4-bit parallel processing. But the unit of its instructions is 8 bits.

Most instructions consist of an 8-bit word and is processed by one machine cycle. But there are some two-word (16 bits) instructions. Also, some require four machine cycles.

The MSM6404 instruction set is similar to those of conventional 4-bit single-chip microcomputers.

Frequently used instructions are one-byt instructions. This makes it possible to use ROM efficiently. Bit operations are available for all the RAM areas, built-in registers and all ports. This makes the instruction set suitable for control use. The provision of multi-function instructions makes it possible to process digits efficiently. Branching of processing can be performed by a conditional skip instruction and jump instruction. These are the major features of the MSM6404 instruction set.

Also provided are a port comparison instruction that makes logic circuit replacement programs (Boolean algebra operations) easy, a push-pop instruction that performs register saving and return collectively, a vertical stacking instruction that makes the multiple entry of subroutines possible, and others.

The MSM6404 provides symmetory for the inter-register transmission instruction, increment instruction and decrement instruction. (There are some exceptions.) That is, bidirectional transmission is possible for registers (including memory and ports) that are available for data transmission. A register having an increment instruction always has a decrement instruction.

For instruction mnemonics (instruction symbols), data transmission is represented by "L" (Load) and the then the initial letter of the destination side is given. (Example: LMA . . . . . M ← A)

There are 121 instructions (the number of mnemonics) which are classified as follows:

(1) Load instructions
Instructions used to transmit data between ACC and athere register (including memory). Including data transmission with built-in counters.

(2) Input/output instructions
Instructions used to input or output the content of ACC to port. Including instructions that output the content of a ROM table to a port.

(3) Operation instructions
Instructions used to perform arrithmetic operations, logical operations and comparison operations between the memory and ACC. Including instructions for port and register comparison, rotation and carry operation.

(4) Exchange instructions

Instructions used to exchange data between the memory and ACC.

(5) Increment and decrement instructions

Instructions that make an increment/decrement of the ACC, memory, H and L.

(6) Bit operation instructions

Instructions used to set, reset and test bits of the memory, ACC, ports and F register.

(7) Branch instructions that change the flow of a program (jump, call, return, etc.).

(8) Interrupt control instructions

Instructions that control the interrupt request flag and interrupt permit/prohibit flag (EIF).

(9) Counter/shift register control instructions

Instructions that control the start/stop of the 8-bit timer/event counter (CT) and shift register (SR).

(10) CPU control instructions

These are NOP, HALT and STOP, which are used to control the operation of the MSM6404.

3.2 INSTRUCTION SYMBOLS

M : Shows the contents of the memory whose address is the contents of the H and L registers.

M(+1) : Shows the content of the memory whose address is the content of the H register and the content of the L register after increment (H, L_{+1}).

Md : Shows the content of the memory whose address is the second byte $I_{7\sim0}$ of the instruction code.

Md(w) : Shows the 8-bit content of the memory whose address is the second byte $I_{7\sim1}$ of the the instruction code.

M(w) : Shows the 8-bit content of the memory whose address is the contents of the H and L registers ($H_{3\sim0}$, $L_{3\sim1}$).

Note: For instructions having the symbol Md(w) or M(w), the lowest bit (L_0, I_0) of the memory address and one word is treated as 8 bits.

P : Shows the I/O port specified by the L register.

Pp : Shows the port specified by the second byte $I_{7\sim4}$ of the instruction code code.

$P(I_1, I_0)$: Shows the bit specified by the immediate values I_1 and I_0 to port P.

$Pp(I_1, I_0)$: Shows the bit specified by I_1 and I_0 of the second byte to the port specified by $I_{7\sim4}$ of the second byte of the instruction code.

I_1	I_0	bit
0	0	0
0	1	1
1	0	2
1	1	3

PC_n : Shows the nth bit of the program counter.

A_n : Shows the nth bit of the accumulator.

I_n : Shows the nth bit of the instruction code.

3.3 LIST OF MSM6404 INSTRUCTIONS

Load instruction push/pop instruction

MNEMONIC	CODE	OPERATION	PAGE
LAI n	9n	A ← n *1	36
LLI n	8n	L ← n *1	43
LHLI nn	15nn	HL ← nn	43
LMI nn	14nn	M(w) ← nn	41
LAL	21	A ← L	38
LLA	2D	L ← A	42
LAH	22	A ← H	38
LHA	2E	H ← A	43
LAM	38	A ← M	36
LMA	2F	M ← A	40
LAM +	24	A ← M, + 1, S	37
LAM -	25	A ← M, - 1, S	36
LMA +	26	M ← A, + 1, S	40
LMA -	27	M ← A, - 1, S	40
LAMM n_2	39 ~ B	A ← M, mod	37
LAMD mm	10mm	A ← Md	37
LMAD mm	11mm	Md ← A	40
LMTD mm	19mm	Md ← T(M(w), A)	42
LMCT	3E59	M(w) ← CT	42
LCTM	3E51	CT ← M(w)	45
LMSR	3E5A	M(w) ← SR	42
LSRM	3E52	SR ← M(w)	45
LTMM	3E50	TM ← (M(w), A)	45
PUSH	1C	STACK ← R *2	38
POP	1D	R ← STACK	39

*1. Vertical stack instruction, *2. R = C, A, H, L

Input/Output instruction

IP	20	A ← P	38
IPD p	3DpD	A ← Pp	37
OP	23	P ← A	44
OPD p	3DpC	Pp ← A	43
OPT	18	$P_{5\cdot4}$ ← T(M(w).A)	44

T = ROM table

Counter/shift register control instruction (PB)

MNEMONIC	CODE	OPERATION	PAGE
ECT	3DBB	CTF ← 1 (start)	69
ESR	3DBA	SRF ← 1 (start)	70
DCT	3DB7	CTF ← 0 (stop)	69
DSR	3DB6	SRF ← 0 (stop)	70
TCT	3DB3	CTF = 1, S	70
TSR	3DB2	SRF = 1, S	71

CPU control instruction (PB)

NOP	00	No Operation	71
HALT	3DB8	Halt CPU	71
STOP	3DB9	Stop Clock	86

Note: S : Instruction with a skip
n : Immediate data
a : ROM address data
m : RAM address data
p : Port address data

Exchange instruction

X	28	A ↔ M	46
XM n_2	29 ~ B	A ↔ M, mod	46
X +	3C	A ↔ M, +1, S	46
X -	2C	A ↔ M, -1, S	46

Increment/decrement instruction

INA	30	A ← A + 1, S	47
INM	33	M ← M + 1, S	47
INL	31	L ← L + 1, S	47
INH	32	H ← H + 1, S	48
INMD mm	12mm	Md ← Md + 1, S	47
DCA	34	A ← A - 1, S	48
DCM	37	M ← M - 1, S	49
DCH	36	H ← H - 1, S	48
DCMD mm	13mm	Md ← Md - 1, S	49

Bit operation instructions

MNEMONIC	CODE	OPERATION	PAGE
TAB n_2	54 ~ 7	A bit (n) = 1, S	55
RAB n_2	64 ~ 7	A bit (n) ← 0	55
SAB n_2	74 ~ 7	A bit (n) ← 1	55
TMB n_2	58 ~ B	M bit (n) = 1, S	56
RMB n_2	68 ~ B	M bit (n) ← 0	55
SMB n_2	78 ~ B	M bit (n) ← 1	56
TFB n_2	5C ~ F	F bit (n) = 1, S	55
RFB n_2	6C ~ F	F bit (n) ← 1	56
SFB n_2	7C ~ F	F bit (n) ← 1	56
TPB n_2	50 ~ 3	P bit (n) = 1, S	54
RPB n_2	60 ~ 3	P bit (n) ← 0	54
SPB n_2	70 ~ 3	P bit (n) ← 1	54
TPBD p.n_2	3D$p_{0\sim3}$	Pp bit (n) = 1, S	56
RPBD p.n_2	3D$p_{4\sim7}$	Pp bit (n) ← 0	57
SPBD p.n_2	3D$p_{8\sim B}$	Pp bit (n) ← 1	57

Interrupt control instructions

MNEMONIC	CODE	OPERATION	PAGE
MEI	3E60	MEIF ← 1	61
MDI	3E61	MEIF ← 0	61
EITB	3DC9	EITBF ← 1	64
EITM	3DCA	EITMF ← 1	65
EICT	3DCB	EICTF ← 1	67
EIEX	3DC8	EIEXF ← 1	62
DITB	3DC5	EITBF ← 0	64
DITM	3DC6	EITMF ← 0	65
DICT	3DC7	EICTF ← 0	67
DIEX	3DC4	EIEXF ← 0	62
TITB	3DC1	EITNF = 1, S	65
TITM	3DC2	EITMF = 1, S	66
TICT	3DC3	EICTF = 1, S	67
TIEX	3DC0	EIEXF = 1, S	62
TQEX	3D20	IRQ EX = 1, S	63
TQTB	3DD0	IRQ TB = 1, S	64
TQTM	3DD1	IRQ TM = 1, S	66
TQCT	3DD2	IRQ CT = 1, S	68
TQSR	3DD3	IRQ SR = 1, S	68
RQEX	3D24	IRQ EX ← 0	63
RQTB	3DD4	IRQ TB ← 0	63
RQTM	3DD5	IRQ TM ← 0	66
RQCT	3DD6	IRQ CT ← 0	68
RQSR	3DD7	IRQ SR ← 0	69

Branch instructions

JCP	a_6	C0 ~ FF	PC ← a_6	58
JP	a_{12}	4 a_{12}	PC ← a_{12}	58
JPL	a_{13}	3F a_{13}	PC ← a_{13} *3	58
CZP	a	B a	CALL 2a	59
CAL	a_{12}	A a_{12}	CALL a_{12}	60
CALL	a_{13}	3F a_{13}	CALL a_{13} *3	60
RT		1 E	Return	60
RTS		1 F	Return S	61
JA		1 A	PC ← (PC ← A) + 1	57
JM		1B	PC ← (M(w), A)	57

*3 Effective only for evaluation chip

Operation/rotation/comparison instructions

ADS		02	A ← A + M, S	49
ADCS		01	A, C ← A+M+C, S	49
ADC		03	A, C ← A+M+C	50
AIS	n	3E 4n	A ← A + n, S	51
DAA		06	A ← A + 6	51
DAS		0A	A ← A + 10	51
AND		0D	A ← A ∧ M	50
OR		05	A ← A ∨ M	50
EOR		04	A ← A ∀ M	50
CMA		0B	A ← $\overline{A}$	50
CIA		0C	A ← $\overline{A}$ + 1	51
RAL		0E	Rotate Left wC	52
RAR		0F	Rotate Right wC	52
TC		09	C = 1 S	52
SC		07	C ← 1	52
RC		08	C ← 0	52
CAI	n	3E 0n	A = n, S	53
CLI	n	3E 2n	L = n, S	53
CPI	p.n	17 pn	Pp = n, S	53
CMI	n	3E 1n	M = n, S	54
CAM		16	A = M S	53

3.4 DESCRIPTION OF INSTRUCTIONS

1. Load and store instructions

LAI n

Instruction code

1	0	0	1	I_3	I_2	I_1	I_0

No. of bytes : 1
Machine cycle : 1
Function : $A \leftarrow I_{3\sim0}$
Explanation : Loads the lower 4 bits of the instruction code to the ACC. If this instruction is written successively, only the first one is executed and the second and subsequent ones become NOP. (vertical stack instruction)
Also, during the execution of this instruction or skipping, and during the LLI instruction, an interrupt is prohibited, through an interrupt request is accepted.

LAM

Instruction code

0	0	1	1	1	0	0	0

No. of bytes : 1
Macnine cycle : 1
Function : $A \leftarrow M$
Explanation : Loads the content of memory M to the ACC. This instruction is equivalent to LAMM 0.

LAM–

Instruction code

0	0	1	0	0	1	0	1

No. of bytes : 1
Machine cycle : 1
Function : $A \leftarrow M$, $L \leftarrow L-1$, Skip if L = F
: Load the content of memory M to the ACC and then performs the decrement of the L register.
If L becomes F, the next instruction is skipped.

No. of bytes : 2
Machine cycle : 2
Function : A ← Md
Explanation : Loads the content of memory Md to the ACC.
Caution : The content of the H and L registers are not changed.

LAMM n2

Instruction code

0	0	1	1	1	0	I_1	I_0

No. of bytes : 1
Machine cycle : 1
Function : A ← M. H ← H ∀ I_1 I_0
Explanation : Loads the content of memory M to the ACC and then modifies (EOR) the content of the H register with the lower two bits of the instruction.

LAM+

Instruction code

0	0	1	0	0	1	0	0

No. of cytes : 1
Machine cycle : 1
Function : A ← M. L ← L + 1, Skip if L = 0
Explanation : Loads the content of memory M to the ACC and then make an increment of the L register. If this results in L = 0, the next instruction is skipped.

IPD p

Instruction code

0	0	1	1	1	1	0	1

P_3	P_2	P_1	P_0	1	1	0	1

No. of bytes : 2
Machine cycle : 2
Function : A ← Pp
Explanation : Loads the content of port Pp to the ACC.
Caution : If external data is applied to port Pp, the data is loaded. Otherwise, the content of the latch of Pp is loaded.

IP

Instruction code

0	0	1	0	0	0	0	0

No. of bytes : 1
Machine cycle : 1
Function : A ← P
Explanation : Loads the content of port P to the ACC.
Caution : If data is applied to port P, the data is loaded. Otherwise, the content of the latch of P is loaded.

LAL

Instruction code

0	0	1	0	0	0	0	1

No. of bytes : 1
Machine cycle : 1
Function : A ← L
Explanation : Loads the content of the L register to the ACC.

LAH

Instruction code

0	0	1	0	0	0	1	0

No. of bytes : 1
Machine cycle : 1
Function : A ← H
Explanation : Loads the content of the H register to the ACC.

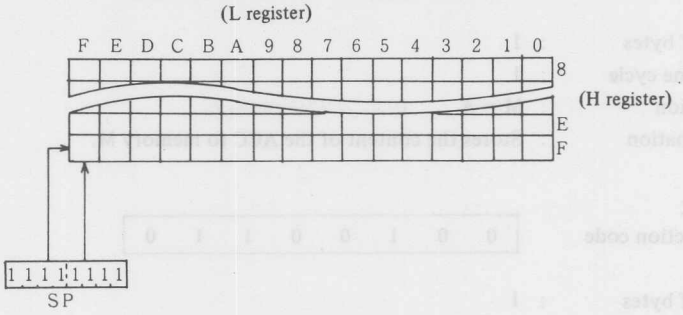
PUSH

Instruction code

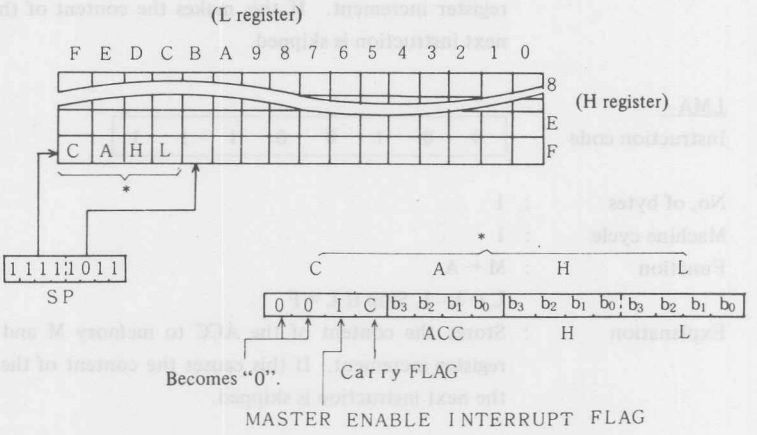
0	0	0	1	1	1	0	0

No. of bytes : 1
Machine cycle : 3
Function : STACK ← C, A, H, L
SP ← SP–4
Explanation : Pushes the register group (C, A, H, L) to the stack. The stack pointer SP is decresed by 4.
Caution : The master enable interrupt flag MEIF is transferred to b1 of the word to which C is saved. However, the MEIF cannot be returned.

Example : STACK before execution of PUSH (Part of RAM)



STACK after execution of PUSH (Part of RAM)



POP

Instruction code

0	0	0	1	1	1	0	1

No. of bytes : 1
Machine cycle : 3
Function : A, C, H, L ← STACK
SP ← SP + 4
Explanation : Returns the register group (C, A, H, L) that has been saved to STACK to individual registers.
The stack pointer SP is increased by 4.

No. of bytes : 1
Machine cycle : 1
Function : M ← A
Explanation : Stores the content of the ACC to memory M.

LMA+

Instruction code

0	0	1	0	0	1	1	0

No. of bytes : 1
Machine cycle : 1
Function : M ← A
Explanation : Stores the content of the ACC to memory M and then makes an L register increment. If this makes the content of the L register 0, the next instruction is skipped.

LMA–

Instruction code

0	0	1	0	0	1	1	1

No. of bytes : 1
Machine cycle : 1
Function : M ← A
L ← L–1, Skip if L = F
Explanation : Stores the content of the ACC to memory M and then makes an L register increment. If this causes the content of the L register to be F, the next instruction is skipped.

LMAD mm

Instruction code

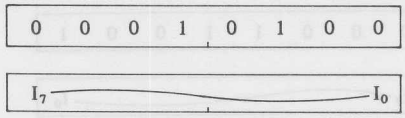
0	0	0	1	0	0	0	1

I_7	I_6	I_5	I_4	I_3	I_2	I_1	I_0

No. of bytes : 2
Machine cycle : 2
Function : Md ← A
Explanation : Stores the content of the ACC to memory Md.

LMI nn

Instruction code



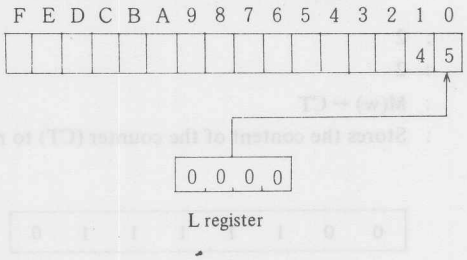
No. of bytes : 2

Machine cycle : 2

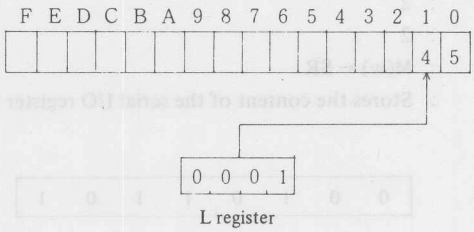
Function : $M(w) \leftarrow I_7 \sim I_0$

Explanation : When bit_0 of the L register is 0, stores the immediate value I_{3-0} to memory M and I_{7-4} to memory M(+1).
If bit_0 of the L register is 1, stores the immediate value I_{7-4} to memory M and I_{3-0} to memory M(−1).
That is, stores the 8 bits of I_{7-0} to the memory addresses shown by H_{3-0} and L_{3-1}. The content of the L register is not changed.

Example : When L_0 is 0, the following is made after execution of LMI 45:



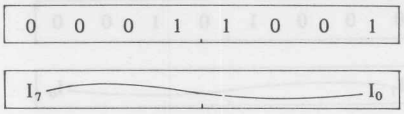
When L_0 is 1, the following is obtained:



This is, bit_0 of the L register becomes an invalid bit.

LMTD mm

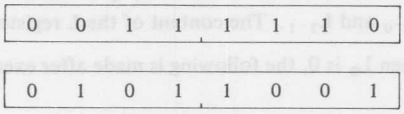
Instruction code



No. of bytes : 2
Machine cycle : 3
Function : Md(w) ← T (M(w), A)
Explanation : Shows the ROM address by a total of the 12 bits of the contents of memory M(w) and the ACC, and stores the contents of the address to memory Md(w).

LMCT

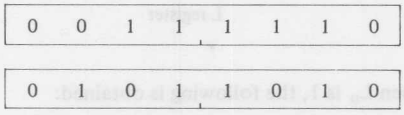
Instruction code



No. of bytes : 2
Machine cycle : 2
Function : M(w) ← CT
Explanation : Stores the content of the counter (CT) to memory M(w).

LMSR

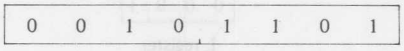
Instruction code



No. of bytes : 2
Machine cycle : 2
Function : M(w) ← SR
Explanation : Stores the content of the serial I/O register SR to memory M(w).

LLA

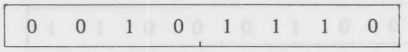
Instruction code



No. of bytes : 1
Machine cycle : 1
Function : L ← A
Explanation : Loads the content of the ACC to the L register.

LHA

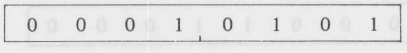
Instruction code

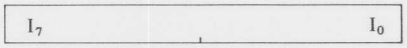


No. of bytes : 1
Machine cycle : 1
Function : H ← A
Explanation : Loads the content of the ACC to the H register.

LHLI nn

Instruction code

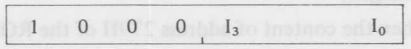




No. of bytes : 2
Machine cycle : 2
Function : HL ← $I_{7\sim4}$ of the second byte of the instruction code to the H register and $I_{3\sim0}$ to the L register.

LLI n

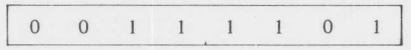
Instruction code

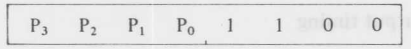


No. of bytes : 1
Machine cycle : 1
Function : L ← $I_{3\sim0}$
Explanation : Loads the immediate value $I_{3\sim0}$ to the L register. If this instruction is placed in the ROM successively, only the first one is executed and skipping is made until the next instruction appears.
Caution : During the execution of this instruction, LAI instruction and skipping, an interrupt is prohibited. However, an interrupt request is accepted.

OPD p

Instruction code





No. of bytes : 2
Machine cycle : 2
Function : Pp ← A
Explanation : Outputs the content of the ACC to port Pp.

OP

Instruction code

0	0	1	0	0	0	1	1

No. of bytes : 1
Machine cycle : 1
Function : P ← A
Explanation : Outputs the content of the ACC to port P.

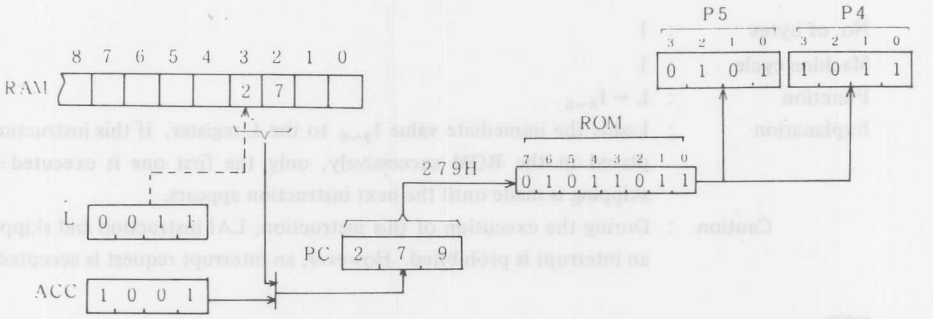
OPT

Instruction code

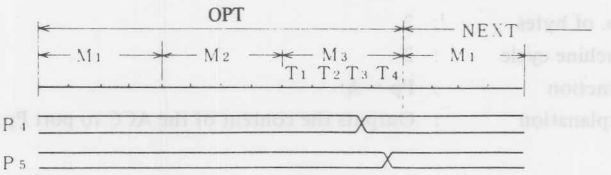
0	0	0	1	1	0	0	0

No. of bytes : 1
Machine cycle : 3
Function : $P_5 P_4 \leftarrow T$ (M(w), A)
Exaplanation : Shows the ROM address by a total of the 12 bits of the 8-bit content of memory M(w) and 4-bit content of the ACC, and outputs the content of the address to ports P_5 and P_4.
The contents of the H and L registers and program counter PC are not changed.

Example 1 : When the content of address 279H of the ROM is output.



Example 2 : Output timing



The lower 4 bits of the ROM is output to port P_4 at T_3 of the third machine cycle. The higher 4 bits are output to port P_5 at T_4.

LTMM

Instruction code

0	0	1	1	1	1	1	0

0	1	0	1	0	0	0	0

No. of bytes : 2
Machine cycle : 2
Function : TM ← (M(w), A)
Explanation : Loads the 12 bit contents of memory M(w) and the ACC to the timer (TM). This value is actually loaded to the 12-bit register and the 12-bit register is reset. (See P31 2.4.3 Programmable Timer.)
Caution : If the value to be loaded is 000H, the timer operation stops.
If this instruction is executed while the timer is in operation by an external asynchronous pulse, it may become identical with the preset value at that point of time.

LCTM

Instruction code

0	0	1	1	1	1	1	0

0	1	0	1	0	0	0	1

No. of bytes : 2
Machine cycle : 2
Function : CT ← M(w)
Explanation : Loads the content of memory M(w) to the counter (CT).
This value is actually loaded to the 8-bit register and the contents of the 8-bit counter is reset. (See P33 2.4.4 Timer/Event Counter.)
Caution : If the value to be loaded is 00H, the count operation stops.

LSRM

Instruction code

0	0	1	1	1	1	1	0

0	1	0	1	0	0	1	0

No. of bytes : 2
Machine cycle : 2
Function : SR ← M(w)
Explanation : Loads the 8-bit contents of memory M(w) to the serial I/O register SR. At this time, the shift counter is reset.

2. Exchange Instructions

X

Instruction code

0	0	1	0	1	0	0	0

No. of bytes : 1
Machine cycle : 1
Function : A ↔ M
Explanation : Exchanges the contents of the ACC and memory M.

XM n2

Instruction code

0	0	1	0	1	0	I_1	I_0

No. of bytes : 1
Machine cycle : 1
Function : A ↔ M. H ← H ∀ I_1 I_0
Explanation : Exchanges the contents of the ACC and memory M. After that, obtains the exclusive logical sum (EOR) of the lower two bits of the H register and immediate $I_1 I_0$ and places the result in the H register.
Caution : When $I_1 I_0$ is 00, this instruction is equivalent to the X instruction.

X+

Instruction code

0	0	1	1	1	1	0	0

No. of bytes : 1
Machine cycle : 1
Function : A ↔ M
L ← L + 1, Skip if L = 0
Explanation : Exchange the contents of the ACC and memory M. After that, make an L register increment. If this causes L = 0, the next instruction is skipped.

X–

Instruction code

0	0	1	0	1	1	0	0

No. of bytes : 1
Machine cycle : 1
Function : A ↔ M
L ← L–1, Skip if L = F
Explanation : Exchanges the contents of the ACC and memory M. After that, makes an L register decrement. If this causes L = F, the next instruction is skipped.

INA

Instruction code

0	0	1	1	0	0	0	0

No. of bytes : 1
Machine cycle : 1
Function : A ← A + 1, Skip if A = 0
Explanaction : Makes an ACC increment. If this causes ACC = 0, the next instruction is skipped.

INM

Instruction code

0	0	1	1	0	0	1	1

No. of bytes : 1
Machine cycle : 1
Function : M ← M + 1, Skip if M = 0
Explanation : Makes an increment of the content of memory M. If this causes M to be 0, the next instruction is skipped.

INMD mm

Instruction code

0	0	0	1	0	0	1	0

I_7		I_0

No. of bytes : 2
Machine cycle : 2
Function : Md ← Md + 1, Skip if Md = 0
Explanation : Makes an increment of the content of memory Md. If this causes Md to be 0, the next instruction is skipped.

INL

Instruction code

0	0	1	1	0	0	0	1

No. of bytes : 1
Machine cycle : 1
Function : L ← L + 1, Skip if L = 0
Explanation : Makes an increment of the content of the L register. If this causes L to be 0, the next instruction is skipped.

INH

Instruction code

0	0	1	1	0	0	1	0

No. of bytes : 1
Machine cycle : 1
Function : H ← H + 1, Skip if H = 0
Explanation : Makes an increment of the content of register H. If this causes H ot be 0, the next instruction is skipped.

DCA

Instruction code

0	0	1	1	0	1	0	0

No. of bytes : 1
Machine cycle : 1
Function : A ← A−1, Skip if A = F
Explanation : Makes a decrement of the content of the ACC. If this results in A = F, the next instruction is skipped.

DCL

Instruction code

0	0	1	1	0	1	0	1

No. of bytes : 1
Machine cycle : 1
Function : L ← L−1, Skip if L = F
Explanation : Makes a decrement of the content of the L register. If this causes L = F, the next instruction is skipped.

DCH

Instruction code

0	0	1	1	0	1	1	0

No. of bytes : 1
Machine cycle : 1
Function : H ← H−1, Skip if H = F
Explanation : Makes a decrement of the content of the H register. If this causes H = F, the next instruction is skipped.

DCMD mm

Instruction code

0	0	0	1	0	0	1	1
I_7							I_0

No. of bytes : 2
Machine cycle : 2
Function : Md ← Md−1, Skip if Md = F
Explanation : Makes a decrement of the content of memory Md. If this causes Md-F, the next instruction is skipped.

DCM

Instruction code

0	0	1	1	0	1	1	1

No. of bytes : 1
Machine cycle : 1
Function : M ← M−1, Skip if M = F
Explanation : Makes a decrement of the content of memory M. If this causes M = F, the next instruction is skipped.

4. **Operation and Comparison Instruction**

ADS

Instruction code

0	0	0	0	0	0	1	0

No. of bytes : 1
Machine cycle : 1
Function : A ← A + M, Skip if Carry = 1
Explanation : Adds the contents of the ACC and memory M in a binary form and places the result in the ACC. At this time, if there is a carry, the next instruction is skipped. Carry flag C does not change.

ADCS

Instruction code

0	0	0	0	0	0	0	1

No. of bytes : 1
Machine cycle : 1
Function : C, A ← A + M + C, Skip if Carry = 1
Explanation : Adds the contents of the ACC, memory M and carry flag C in a binary form and places the result in the ACC and carry flag C. If there is a carry, the next instruction is skipped.

ADC

Instruction code

0	0	0	0	0	0	1	1

No. of bytes : 1
Machine cycle : 1
Function : C, A ← A + M + C
Explanation : Adds the ACC, memory M and carry flag C in a binary form and places the result in the ACC and C.

AND

Instruction code

0	0	0	0	1	1	0	1

No. of bytes : 1
Machine cycle : 1
Function : A ← A ∧ M
Explanation : Obtains AND (logical product) of the contents of the ACC and memory M and places the result in the ACC.

OR

Instruction code

0	0	0	0	0	1	0	1

No. of bytes : 1
Machine cycle : 1
Function : A ← A ∨ M
Explanation : Obtains OR (logical sum) of the contents of the ACC and memory M and places the result in the ACC.

EOR

Instruction code

0	0	0	0	0	1	0	0

No. of bytes : 1
Machine cycle : 1
Function : A ← A ∀ M
Explanation : Obtains EOR (exclusive logical sum) of the contents of the ACC and memory M and places the result in the ACC.

CMA

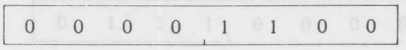
Instruction code

0	0	0	0	1	0	1	1

No. of bytes : 1
Machine cylce : 1
Function : A ← $\overline{A}$
Explanation : Places the value that is the reversal of the content of the ACC (complement of 1) in the ACC.

CIA

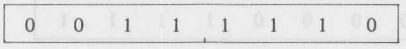
Instruction code

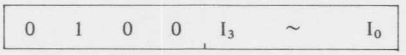


No. of bytes : 1
Machine cycle : 1
Function : $A \leftarrow \overline{A} + 1$
Explanation : Reverses the content of the ACC and places the value with a added (complement of 2) in the ACC.

AIS n

Instruction code

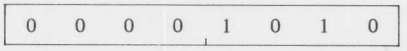




No. of bytes : 2
Machine cycle : 2
Function : $A \leftarrow A + I_{3\sim 0}$, Skip if Carry = 1
Explanation : Adds the content of the ACC and immediate value 13–14 in a binary form and places the result in the ACC. At this time, if there is a carry, the next instruction is skipped.

DAS

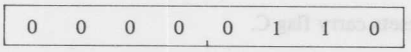
Instruction code



No. of bytes : 1
Machine cycle : 1
Function : $A \leftarrow A + 10$
Explanation : Adds 10 (A in hexadecimal notation) to the content of the ACC. This instruction is used for decomal correction of the content of the ACC in a deciaml subtraction program.

DAA

Instruction code



No. of bytes : 1
Machine cycle : 1
Function : $A \leftarrow A + 6$
Explanation : Adds 6 to the content of the ACC. This instruction is used for decimal correction of the content of the ACC in a decimal addition program.

No. of bytes : 1
Machine cycle : 1
Function : $C \leftarrow A_3$, $A_n \leftarrow A_{n-1}$ (m = 1 ~ 3), $A_0 \leftarrow C$
Explanation : Rotates the contents of the ACC including carry flag C by one bit to the left.

RAR

Instruction code

0	0	0	0	1	1	1	1

No. of bytes : 1
Machine cycle : 1
Function : $C \rightarrow A_3$, $A_n \rightarrow A_{n-1}$ (n = 1~ 3), $A_0 \rightarrow C$
Explanation : Rotates the content of the ACC including carry flag C by one bit to the right.

SC

Instruction code

0	0	0	0	0	1	1	1

No. of bytes : 1
Machine cycle : 1
Function : $C \leftarrow 1$
Explanation : Sets carry flag C.

RC

Instruction code

0	0	0	0	1	0	0	0

No. of bytes : 1
Machine cycle : 1
Function : $C \leftarrow 0$
Explanation : Resets carry flag C.

TC

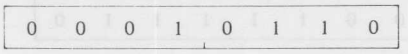
Instruction code

0	0	0	0	1	0	0	1

No. of bytes : 1
Machine cycle : 1
Function : Skip if C = 1
Explanation : Tests carry flag C. If it is "1", the next instruction is skipped.

CAM

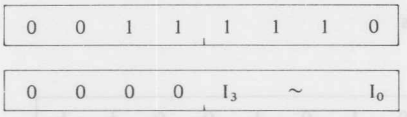
Instruction code



No. of bytes : 1
Machine cycle : 1
Function : Skip if A = M
Explanation : Compares the contents of the ACC and memory M. If these are equal, the next instruction is skipped. The contents of the ACC, memory and carry flag are not changed.

CAI n

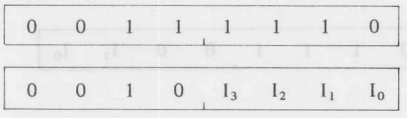
Instruction code



No. of bytes : 2
Machine cycle : 2
Function : Skip if A = $I_{3\sim0}$
Explanation : Compares the contents of the ACC and immediate value $I_{3\sim0}$. If these are equal, the next instruction is skipped.

CLI n

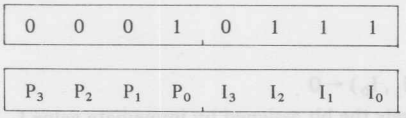
Instruction code



No. of bytes : 2
Machine cycle : 2
Function : Skip if L = $I_{3\sim0}$
Explanation : Compares the L register and immediate value $I_{3\sim0}$. If these are equal, the next instruction is skipped.

CPI p, n

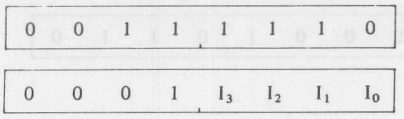
Instruction code



No. of bytes : 2
Machine cycle : 2
Function : Skip if Pp = $I_{3\sim0}$
Explanation : Compares the content of the port (Pp) specified by $P_{3\sim0}$ and immediate value $I_{3\sim0}$. If these are equal, the next instruction is skipped.
Caution : If no input signal is applied to port Pp, the output data of the port is compared.

CMI n

Instruction code

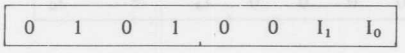


No. of bytes : 2
Machine cycle : 2
Function : Skip if $M = I_{3 \sim 0}$
Explanation : Compares the content of memory M and immediate value $I_{3 \sim 0}$. If these are equal, the next instruction is skipped.

5. Bit Operation Instructions

TPB n_2

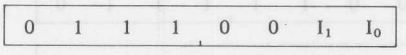
Instruction code



No. of bytes : 1
Machine cycle : 1
Function : Skip if $P(I_1, I_0) = 1$
Explanation : Tests the bit assigned by immediate value I_1 I_0 to port P. If it is "1", the next instruction is skipped.

SPB n_2

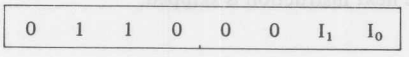
Instruction code



No. of bytes : 1
Machine cycle : 1
Function : $P(I_1, I_0) \leftarrow 1$
Explanation : Sets the bit assigned by immediate value I_1 I_0 to port P.

RPB n_2

Instruction code



No. of bytes : 1
Machine cycle : 1
Function : $P(I_1, I_0) \leftarrow 0$
Explanation : Resets the bit assigned by immediate value I_1 I_0 to port P.

TAB n_2

Instruction code

0	1	0	1	0	1	I_1	I_0

No. of bytes : 1
Machine cycle : 1
Function : Skip if A (I_1, I_0) = 1
Explanation : Tests the bit assigned by immediate value I_1 I_0 to the ACC. If it is "1", the next instruction is skipped.

SAB n_2

Instruction code

0	1	1	1	0	1	I_1	I_0

No. of bytes : 1
Machine cycle : 1
Function : A (I_1, I_0) ← 1
Explanation : Sets the bit assigned by immediate value I_1 I_0 to the ACC.

RAB n_2

Instruction code

0	1	1	0	0	1	I_1	I_0

No. of bytes : 1
Machine cycle : 1
Function : A (I_1, I_0) ← 0
Explanation : Resets the bit assigned by immediate value I_1 I_0 to the ACC.

RMB n_2

Instruction code

0	1	1	0	1	0	I_1	I_0

No. of bytes : 1
Machine cycle : 1
Function : M (I_1, I_0) ← 0
Explanation : Resets the bit specified by immediate value I_1 I_0 to memory M.

TFB n_2

Instruction code

0	1	0	1	1	1	I_1	I_0

No. of bytes : 1
Machine cycle : 1
Function : Skip if F (I_1, I_0) = 1
Explanation : Tests the bit assigned by immediate value I_1 I_0 to the F register. If it is "1", the next instruction is skipped.

TMB n_2

Instruction code

0	1	0	1	1	0	I_1	I_0

No. of bytes : 1
Machine cycle : 1
Function : Skip if M (I_1, I_0) = 1
Explanation : Tests the bit assigned by immediate value I_1 I_0 to memory M. If it is "1", the next instruction is skipped.

SMB n_2

Instruction code

0	1	1	1	1	0	I_1	I_0

No. of bytes : 1
Machine cycle : 1
Function : M (I_1, I_0) ← 1
Explanation : Sets the bit assigned by immediate value I_1 I_0 to memory M.

SFB n_2

Instruction code

0	1	1	1	1	1	I_1	I_0

No. of bytes : 1
Machine cycle : 1
Function : F (I_1, I_0) ← 1
Explanation : Sets the bit assigned by immediate value I_1 I_0 to the F register.

RFB n_2

Instruction code

0	1	1	0	1	1	I_1	I_0

No. of bytes : 1
Machine cycle : 1
Function : F (I_1, I_0) ← 0
Explanation : Resets the bit assigned by immediate value I_1 I_0 to the F register.

TPBD p, n_2

Instruction code

0	0	1	1	1	1	0	1

P_3	P_2	P_1	P_0	0	0	I_1	I_0

No. of bytes : 2
Machine cycle : 2
Function : Skip if Pp (I_1, I_0) = 1
Explanation : Tests the bit assigned by immediate value I_1 I_0 to port Pp. If it is "1", the next instruction is skipped.

SPBD p, n_2

Instruction code

0	0	1	1	1	1	0	1
P_3	P_2	P_1	P_0	1	0	I_1	I_0

No. of bytes : 2
Machine cycle : 2
Function : Pp (I_1, I_0) ← 1
Explanation : Sets the bit assigned by immediate value I_1 I_0 to port Pp.

RPBD p, n_2

Instruction code

0	0	1	1	1	1	0	1
P_3	P_2	P_1	P_0	0	1	I_1	I_0

No. of bytes : 2
Machine cycle : 2
Function : Pp (I_1, I_0) ← 0
Explanation : Resets the bit assigned by immediate value I_1 I_0 to port Pp.

6. Branch Instructions

JA

Instruction code

0	0	0	1	1	0	1	0

No. of bytes : 1
Machine cycle : 2
Function : $PC_{3\sim0}$ ← A, then PC + 1
Explanation : Loads the content of the ACC to the lower 4 bits of program counter PC and makes an increment of PC.
Therefore, 16 branches are made depending on the content of the ACC.
Caution : The destination of jumping may be reversed depending on the address at which the instruction is placed.

JM

Instruction code

0	0	0	1	1	0	1	1

No. of bytes : 1
Machine cycle : 2
Function : $PC_{3\sim0}$ ← A. $PC_{7\sim4}$ ← M. $PC_{11\sim8}$ ← M(+1)
Explanation : Loads the 12 bits of the contents of memory M(+1), M and ACC to the program counter.
Therefore, jumping is available over the entire ROM address area.

JCP

Instruction code

1	1	I_5	I_4	I_3	I_2	I_1	I_0

No. of bytes : 1
Machine cycle : 1
Function : $PC_{5\sim0} \leftarrow I_{5\sim0}$
Explanation : Jumps to the address specified by immediate value $I_{5\sim0}$. If this instruction is placed at the end of the page, jumping is made within the page.

JP a12

Instruction code

0	1	0	0	I_{11}	I_{10}	I_9	I_8

I_7	I_6	I_5	I_4	I_3	I_2	I_1	I_0

No. of bytes : 2
Machine cycle : 2
Function : $PC_{11\sim0} \leftarrow I_{11} \sim I_0$
Explanation : Loads immediate value $I_{11\sim0}$ to program counter $PC_{11\sim0}$. That is, jumping can be made to the 4K-byte area of the ROM address.

JPL a13

Instruction code

0	0	1	1	1	1	1	1

0	X	X	I_{12}	I_{11}	I_{10}	I_9	I_8

I_7	I_6	I_5	I_4	I_3	I_2	I_1	I_0

No. of bytes : 3
Machine cycle : 4
Function : $PC_{12\sim0} \leftarrow I_{12\sim0}$
Explanation : Loads immediate value to $PC_{12\sim0}$. That is, jumping can be made to the entire 8K-byte area of the ROM address.
Caution : Bit 6 and bit 5 of the second byte are invalid bid and may to "1" or "0". This instruction is effective only for the evaluation chip.

CZP a

Instruction code

1	0	1	1	I_3	I_2	I_1	I_0

No. of bytes : 1

Machine cycle : 4

Function : $STACK \leftarrow PC + 1$

$PC_{12 \sim 5, 0} \leftarrow 0, PC_{4 \sim 1} \leftarrow I_{3 \sim 0}$

$SP \leftarrow SP-4$

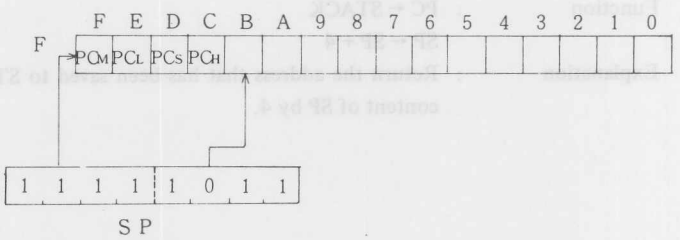
Explanation : Loads immediate value $I_{3 \sim 0}$ to $PC_{4 \sim 1}$ and "0" to $PC_{12 \sim 5}$,) after saving the content of PC to STACK.

Decrease stack pointer SP by 4. That is, the 16 addresses of the page 0 area of the ROM every two addresses by immediate value $I_{3 \sim 0}$.

Example 1 : Correspondence between immediate value and call address

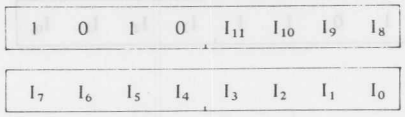
$I_3 \sim I_0$	ADDRESS
0	0 0 0 H
1	0 0 2 H
2	0 0 4 H
3	0 0 6 H
4	0 0 8 H
5	0 0 A H
6	0 0 C H
7	0 0 E H
8	0 1 0 H
9	0 1 2 H
A	0 1 4 H
B	0 1 6 H
C	0 1 8 H
D	0 1 A H
E	0 1 C H
F	0 1 E H

Example 2 : Saving to the stack (After execution of CZP)



CAL a12

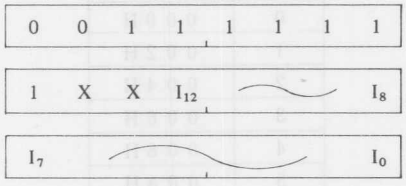
Instruction code



No. of bytes : 2
Machine cycle : 4
Function : STACK ← PC + 2
$PC \leftarrow I_{11 \sim 0}$
SP ← SP–4
Explanation : Saves the content of PC to STACK and then loads immediate value $I_{11 \sim 0}$ to $PC_{11 \sim 0}$. Decreases stack pointer SP by 4. That is, it is possible to call the 4K-byte area of the ROM address.

CALL a13

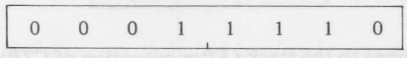
Instruction code



No. of bytes : 3
Machine cycle : 4
Function : STACK ← PC + 3
$PC_{12 \sim 0} \leftarrow I_{12 \sim 0}$
SP ← SP–4
Explanation : Saves the content of PC to STACK and loads immediate value $I_{12 \sim 0}$ to $PC_{12 \sim 0}$. That is, it is possible to call the 8K-byte area of the ROM address. This instruction is effective only for the evaluation chip.

RT

Instruction code



No. of bytes : 1
Machine cycle : 4
Function : PC ← STACK
SP ← SP + 4
Explanation : Return the address that has been saved to STACK to PC. Increase the content of SP by 4.

RTS

Instruction code

0	0	0	1	1	1	1	1

No. of bytes : 1
Machine cycle : 4
Function : PC ← STACK, SP ← SP + 4
Skip Unconditional
Explanation : Return the address that has been saved to STACK to PC and skips the instruction in that address.
Increase the content of SP by 4.

7. **Interrupt Control Instruction**

MEI

Instruction code

0	0	1	1	1	1	1	0

0	1	1	0	0	0	0	0

No. of bytes : 2
Machine cycle : 2
Function : MEIF ← 1
Explanation : Sets master enable interrupt falg MEIF to allow an interrupt. The interrupt is allowed after the execution of the next instruction.
If an interrupt occurs, the flag is reset and put to an MDI state. In an MEI state, the execution of individual EI and DI instructions is invalid.

MDI

Instruction code

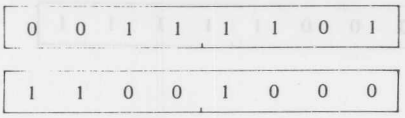
0	0	1	1	1	1	1	0

0	1	1	0	0	0	0	1

No. of bytes : 2
Machine cycle : 2
Function : MEIF ← 0
Explanation : Resets master enable interrupt flag MEIF to prohibit an interrupt.

EIEX

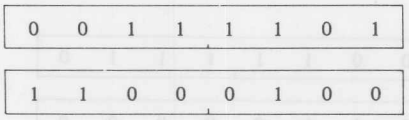
Instruction code



No. of bytes : 2
Machine cycle : 2
Function : EIEXF ← 1. (PC bit_0 ← 1)
Explanation : Sets external interrupt enable flag EIEXF to enable an external interrupt. The interrupt is actually available after the MEI flag is set.
Caution : The execution of this instruction is invalid (is not set) if MEIF is 1.

DIEX

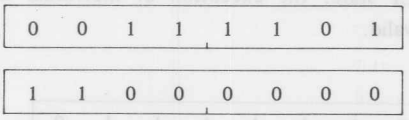
Instruction code



No. of bytes : 2
Machine cycle : 2
Function : EIEXF ← 0, (PC bit_0 ← 0)
Explanation : Resets external interrupt enable flag EIEXF to prohibit an external interrupt.
Caution : The execution of this instruction is invalid (is not reset) if MEIF is 1.

TIEX

Instruction code



No. of bytes : 2
Machine cycle : 2
Explanation : Tests external interrupt enable falg EIEXF. If it is 1, skips the next instruction. If it is 0, executes the next instruction.
The content of EIEXF does not change.

RQEX

Instruction code

0	0	1	1	1	1	0	1

0	0	1	0	0	1	0	0

No. of bytes : 2
Machine cycle : 2
Function : IRQEXF ← 0, (P2 bit_1 ← 0)
Explanation : Resets external interrupt request flag IRQEXF.
Caution : If the execution of this instruction is simultanous with the fall of the input signal of P20 (external interrupt signal), priority is given to this instruction (reset) and the input signal is ignored.

TQEX

Instruction code

0	0	1	1	1	1	0	1

0	0	1	0	0	0	0	0

No. of bytes : 2
Machine cycle : 2
Function : Skip if IRQEXF = 1, (Skip if P2 bit_0 = 1)
Explanation : Tests external interrupt request flag IRQEXF. If it is 1, skips the next instruction. If it is 0, executes the next instruction.
The content of IRQEXF is not changed.

RQTB

Instruction code

0	0	1	1	1	1	0	1

1	1	0	1	0	1	0	0

No. of bytes : 2
Machine cycle : 2
Function : IRQTBF ← 0, (PD bit_0 ← 0)
Explanation : Resets time base interrupt request flag IRQTBF.
Caution : If the execution of this instruction and the time base interrupt request are simultaneous, priority is given to this instruction (reset), and the time base interrupt request is ignored.

TQTB

Instruction code

0	0	1	1	1	1	0	1
1	1	0	1	0	0	0	0

No. of bytes : 2
Machine cycle : 2
Function : Skip if IRQTBF = 1, (Skip if PD bit_0 = 1)
Explanation : Tests time base interrupt request flag IRQTBF. If it is 1, skips the next instruction. If it is 0, execute the next instruction.
The content of IRQTBF is not changed.

EITB

Instruction code

0	0	1	1	1	1	0	1
1	1	0	0	1	0	0	1

No. of bytes : 2
Machine cycle : 2
Function : EITBF ← 1, (PC bit 1 ← 1)
Explanation : Sets time base interrupt enable flag EITBF to permit a time base interrupt. However, the interrupt actually becomes available after the MEI flag is set.
Caution : The execution of this instruction is invalid (is not set) if MEIF is 1 (master interrupt enable).

DITB

Instruction code

0	0	1	1	1	1	0	1
1	1	0	0	0	1	0	1

No. of bytes : 2
Machine cycle : 2
Function : EITBF ← 0, (PC bit 1 ← 0)
Explanation : Resets time base interrupt enable flag EITBF and prohibits a time base interrupt.
Caution : The execution of this instruction is invalid (is not reset) if MEIF is 1 (master interrupt enable).

1	1	0	0	0	0	0	1

No. of bytes : 2
Machine cycle : 2
Function : Skip if EITBF = 1, (Skip if PC bit_1 = 1)
Explanation : Tests time base interrupt flag EITBF. If it is 1, skips the next instruction. The content of EITBF is not changed.

EITM

Instruction code

0	0	1	1	1	1	0	1

1	1	0	0	1	0	1	0

No. of bytes : 2
Machine cycle : 2
Function : EITMF ← 1, (PC bit_2 ← 1)
Explanation : Sets timer interrupt enable flag EITMF to enable an timer interrupt. However, the interrupt actually becomes available after the MEI flag is set.
Caution : The execution of this instruction is invalid (is not set) if MEIF is 1.

DITM

Instruction code

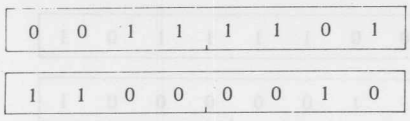
0	0	1	1	1	1	0	1

1	1	0	0	0	1	1	0

No. of bytes : 2
Machine cycle : 2
Function : EITMF ← 0
Explanation : Resets timer interrupt enable flag EITMF to prohibit a timer interrupt.
Caution : The execution of this instruction is invalid (is not reset) if MEIF is 1.

TITM

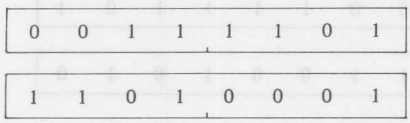
Instruction code



No. of bytes : 2
Machine cycle : 2
Function : Skip if EITMF = 1, (Skip if PC bit_2 = 1)
Explanation : Tests timer interrupt enable falg EITMF. If it is 1, skips the next instruction. Otherwise, executes the next instruction. The content of EITMF is not changed.

TQTM

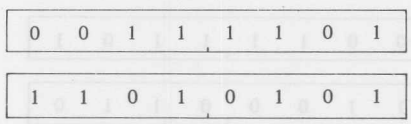
Instruction code



No. of bytes : 2
Machine cycle : 2
Function : Skip if IRQTMF = 1, (Skip if PD bit_1 = 1)
Explanation : Tests timer interrupt request flag IRQTMF. If it is 1, skips the next instruction. Otherwise, executes the next instruction.
The content of IRQTMF is not changed.

RQTM

Instruction code



No. of bytes : 2
Machine cycle : 2
Function : IRQTMF ← 0, (PD bit_1 ← 0)
Explanation : Resets timer interrupt request flag IRQTMF.
Caution : If the execution of this instruction and the timer interrupt request is simultaneous, priority is given to this instruction (reset), and the time base interrupt request is ignored.

EICT

Instruction coe

0	0	1	1	1	1	0	1

1	1	0	0	1	0	1	1

No. of bytes : 2
Machine cycle : 2
Function : EICTF ← 1, (PC bit_3 ← 1)
Explanation : Sets counter interrupt and shift register interrupt enable flag EICTF to permit the two interrupts.
However, these interrupts actually become available after the MEI flag is set.
Caution : The execution of this instruction is invalid (is not set) is MEIF is 1.

DICT

Instruction code

0	0	1	1	1	1	0	1

1	1	0	0	0	1	1	1

No. of bytes : 2
Machine cycle : 2
Function : EICTF ← 0, (PC bit_3 ← 0)
Explanation : Resets counter interrupt and shift register interrupt enable flag EICTF to prohibit the two interrupts.
Caution : The execution of this instruction is invalid (is not reset) if MEIF is 1.

TICT

Instruction code

0	0	1	1	1	1	0	1

1	1	0	0	0	0	1	1

No. of bytes : 2
Machine cycle : 2
Function : Skip if EICTF = 1, (Skip if PC bit_3 = 1)
Explanation : Tests counter interrupt enable flag EICTF. If it is 1, skips the next instruction. Otherwise, executes the next instruction. The content of EICTF is not changed.

TQCT

Instruction code

0	0	1	1	1	1	0	1

1	1	0	1	0	0	1	0

No. of bytes : 2
Machine cycle : 2
Function : Skip if IRQCTF = 1, (Skip if PD bit_2 = 1)
Explanation : Tests counter interrupt request flag IRQCTF. If it is 1, skips the next instruction. Otherwise, executes the next instruction.
The content of IRQCTF is not changed.

RQCT

Instruction code

0	0	1	1	1	1	0	1

1	1	0	1	0	1	1	0

No. of bytes : 2
Machine cycle : 2
Function : IRCCTF ← 0, (PD bit_2 ← 0)
Explanation : Resets counter interrupt request flag IRQCTF.
Caution : If the execution of this instruction an the counter interrupt request are simultanous, priority is given to this instruction (reset), and the counter interrupt request is ignored.

TQSR

Instruction code

0	0	1	1	1	1	0	1

1	1	0	1	0	0	1	1

No. of bytes : 2
Machine cycle : 2
Function : Skip if IRQSRF = 1, (Skip if PD bit_3 = 1)
Explanation : Tests shift register interrupt request flag IRQSRF. If it is 1, skips the next instruction. Otherwise, executes the next instruction.
The content of IRQSRF is not changed.

1	1	0	1	0	1	1	1

No. of bytes : 2
Machine cycle : 2
Function : IRQSRF ← 0, (PD bit_3 ← 0)
Explanation : Resets shift register interrupt request flag IRQSRF.
Caution : If the execution of this instruction and the ounter interrupt request occur simultanously, priority is given to this instruction (reset), and the shift register interrupt request is ignored.

8. Counter/shift Register Control Instructions

ECT

Instruction code

0	0	1	1	1	1	0	1

1	0	1	1	1	0	1	1

No. of bytes : 2
Machine cycle : 2
Function : CTF ← 1 (PB bit_3 ← 1)
Explanation : Enables the timer/event counter (CT). In this state, the timer/event counter can ount up CIN (P10) pulses or internal pulses.

DCT

Instruction code

0	0	1	1	1	1	0	1

1	0	1	1	0	1	0	1

No. of bytes : 2
Machine cycle : 2
Function : CTF ← 0 (PB bit_3 ← 0)
Explanation : Disables the timer/event counter (CT). In this state, the timer/event counter does not count up CIN or internal counter pulses, if any.

TCT

Instruction code

0	0	1	1	1	1	0	1

1	0	1	1	0	1	1	1

No. of bytes : 2
Machine cycle : 2
Function : Skip if CTF = 1
Explanation : Tests counter control flag CTF. If it is 1, skips the next instruction. Otherwise, executes the next instruction.
The contents of CTF is not changed.

ESR

Instruction code

0	0	1	1	1	1	0	1

1	0	1	1	1	0	1	0

No. of bytes : 2
Machine cycle : 2
Function : SRF ← 1, (PB bit_2 ← 1)
Explanation : Enables the shift register (SR). In this stage, the shift register can be operated.
If SRF is 1, that is, when the shift register is in operation, the LSRM instruction should not be executed.
Caution : If the ESR instruction is executed sith SCK (P01) being 0, the clock is applied to the shift register and one bit is shifted. It is, therefore, necessary to execute the ESR instruction when SCK is 1. (See P35 2.4.5 Shift Register.)

DSR

Instruction code

0	0	1	1	1	1	0	1

1	0	1	1	0	1	1	0

No. of bytes : 2
Machine cycle : 2
Function : SRF ← 0, (PB bit_2 ← 0)
Explanation : Disables the shift register (SR). In this state, neither the shift register nor the shift counter is operated even if a serial clock is given.
The LSRM instruction can be executed only when SRF is 0.

TSR

Instruction code

0	0	1	1	1	1	0	1	
1	0	1	1	0	0	1	0	

No. of bytes : 2
Machine cycle : 2
Function : Skip if SRF = 1, (Skip if PB bit_2 = 1)
Explanation : Tests shift register control flag SRF. If it is 1, skips the next instruction. Otherwise, executes the next instruction. The content of SRF is not changed.

9. CPU Control Instruction

NOP

Instruction code

0	0	0	0	0	0	0	0	

No. of bytes : 1
Machine cycle : 1
Function : No Operation
Explanation : Performs nothing. Only the time of one machine cycle is used.

HALT

Instruction code

0	0	1	1	1	1	0	1	
1	0	1	1	1	0	0	0	

No. of bytes : 2
Machine cycle : 2
Function : Halt CPU (PB $bit_0 \leftarrow 1$)
Explanation : Halts the program execution of the CPU. In a halt state, only the program counter stops. No internal function stops.
A halt state is cleared by an interrupt or reset input. In the case of reset input, execution restarts from address 0. In the case of an interrupt, it restarts from the address of each interrupt. If HALT is executed in an MDI state, execution restarts with the instruction next to the HALT by an interrupt request with EIF = 1.
If HALT is executed with MEIF and all EIF are in a "0" (MDI, DI) state, the halt state can be cleared only by resetting.

Caution : Since bit_0 of PB on the hardware is a HALT flag, it is possible to execute HALT by setting the flag by an equivalent instruction such as SPB_{n2}, OP, or OPDp.

STOP

Instruction code

0	0	1	1	1	1	0	1	
1	0	1	1	1	0	0	1	

No. of bytes : 2

Machine cycle : 2

Function : Stop System Clock (PB $bit_1 \leftarrow 1$)

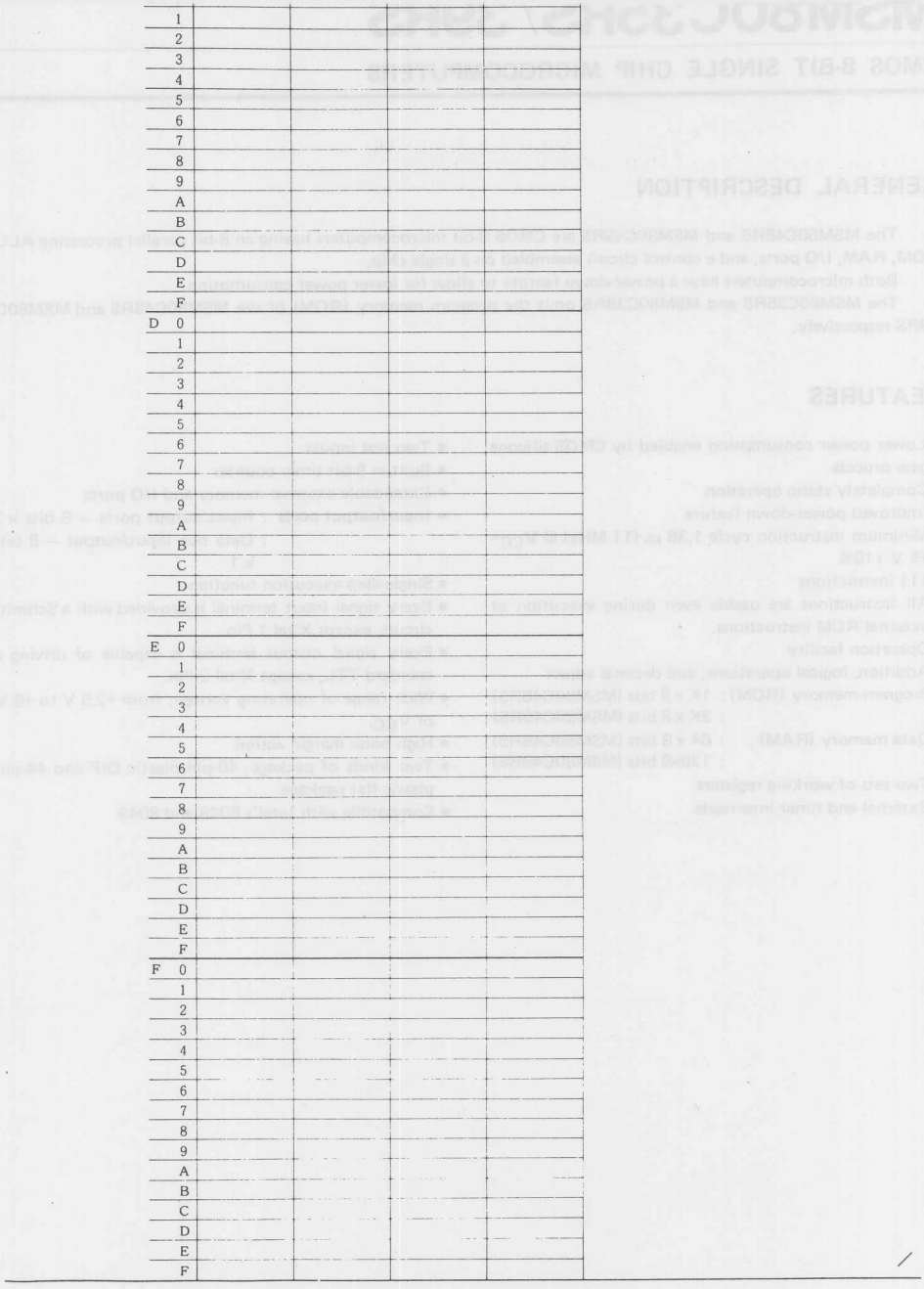
Explanation : Stops the oscillation of the system clock. All the operations of the chip including the CPU stop, and power consumption is minimized.

However, the 12-bit timer and shift register can be operated by an external clock.

A stop state can be cleared by a P20 external interrupt, a timer interrupt, or reset input. At this time, an input pulse which is longer than the time required form the restart of oscillation until its stabilization is necessary. (See P18 2.2.8.)

This pulse should not have chattering. When resetting an interrupt, execution restarts with the next instruction if an MDI state exists. Otherwise, an interrupt is accepted.

Caution : Since bit_1 of PB on the hardware is the STOP flag, it is possible to execute STOP by setting the flag by an equivalent instruction such as $SPB_{n\,2}$, OP, or OPDp.



OKI semiconductor

MSM80C48RS/49RS
MSM80C35RS/39RS

CMOS 8-BIT SINGLE CHIP MICROCOMPUTERS

GENERAL DESCRIPTION

The MSM80C48RS and MSM80C49RS are CMOS 8-bit microcomputers having an 8-bit parallel processing ALU, ROM, RAM, I/O ports, and a control circuit assembled on a single chip.

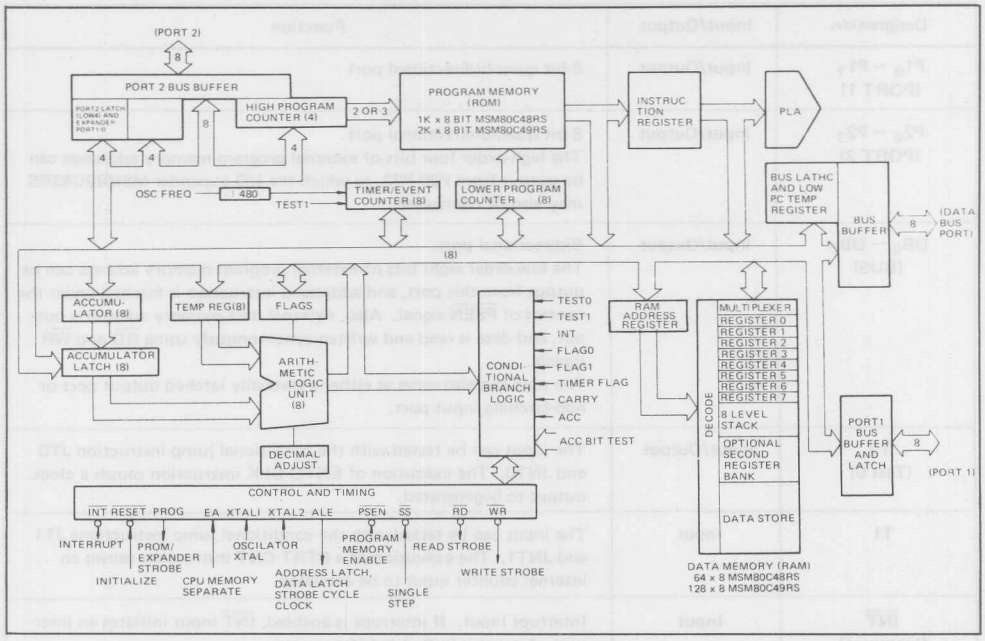
Both microcomputers have a power-down feature to allow for lower power consumption.

The MSM80C35RS and MSM80C39RS omit the program memory (ROM) of the MSM80C48RS and MSM80C-49RS respectively.

FEATURES

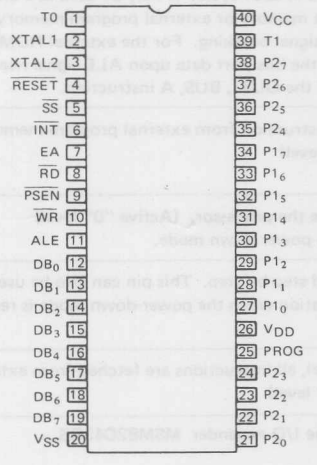
- Lower power consumption enabled by CMOS silicone gate process
- Completely static operation
- Improved power-down feature
- Minimum instruction cycle 1.36 μs (11 MHz) @ V_{CC}= +5 V ±10%
- 111 instructions
- All instructions are usable even during execution of external ROM instructions.
- Operation facility
  Addition, logical operations, and decimal adjust
- Program memory (ROM) : 1K x 8 bits (MSM80C48RS)
  : 2K x 8 bits (MSM80C49RS)
- Data memory (RAM) : 64 x 8 bits (MSM80C48RS)
  : 128x8 bits (MSM80C49RS)
- Two sets of working registers
- External and timer interrupts
- Two test inputs
- Built-in 8-bit timer counter
- Extendable external memory and I/O ports
- Input/output ports : Input/output ports — 8 bits x 2
  : Data bus input/output — 8 bits x 1
- Single-step execution function
- Every signal input terminal is provided with a Schmitt circuit, except X'tal 1 Pin.
- Every signal output terminal is capable of driving a standard TTL, except X'tal 2 Pin.
- Wide range of operating voltage, from +2.5 V to +6 V of V_{CC}.
- High noise margin action
- Two kinds of package; 40-pin plastic DIP and 44-pin plastic flat package
- Compatible with Intel's 8048 and 8049

FUNCTIONAL BLOCK DIAGRAM



PIN CONFIGURATION

(Top View) 40 Lead Plastic DIP



Pin Name	
$P1_0 \sim P1_7$	: Input/output port (PORT1)
$P2_0 \sim P2_7$	: Input/output port (PORT2)
$DB_0 \sim DB_7$	: Data bus port (BUS PORT)
T0, T1	: Test
$\overline{INT}$	: Interrupt
$\overline{RD}$	: Read
$\overline{WR}$	: Write
ALE	: Address Latch Enable
$\overline{PSEN}$	: Program Store Enable
$\overline{RESET}$	: Reset
$\overline{SS}$	: Single Step
EA	: ROM Mode
XTAL 1, 2	: Crystal Controlled Oscillation

$P1_0 \sim P1_7$ (PORT 1)	Input/Output	8-bit quasi-bidirectional port
$P2_0 \sim P2_7$ (PORT 2)	Input/Output	8-bit quasi-bidirectional port The high-order four bits of external program memory addresses can be output from P20-P23, to which the I/O expander MSM82C43RS may also be connected.
$DB_0 \sim DB_7$ (BUS)	Input/Output	Bidirectional port. The low-order eight bits of external program memory address can be output from this port, and addressed instruction is fetched under the control of $\overline{PSEN}$ signal. Also, external data memory address is output, and data is read and written synchronously using $\overline{RD}$ and $\overline{WR}$ signals. The port can also serve as either a statically latched output port or non-latching input port.
T0 (Test 0)	Input/Output	The input can be tested with the conditional jump instruction JT0 and JNT0. The execution of ENTO CLK instruction causes a clock output to begenerated.
T1	Input	The input can be tested with the conditional jump instructions JT1 and JNT1. The execution of a STRT CNT instruction causes an internal counter input to be activated.
$\overline{INT}$ (Interrupt)	Input	Interrupt input. If interrupt is enabled, $\overline{INT}$ input initiates an interrupt. Interrupt is disabled after a reset. Also testable with a JNI instruction. Used to terminate the power-down mode. (Active "0" level)
$\overline{RD}$ (Read)	Output	A signal to read data from external data memory is output. (Active "0" level)
$\overline{WR}$ (Write)		A signal to write data to external data memory is output. (Active "0" level)
ALE (Address & Data Latch Clock)		This signal is generated in each cycle. It may be used as a clock output. External data memory or external program memory is addressed upon ALE signal breaking. For the external ROM, this signal is used to latch the bus port data upon ALE signal rise-up after the execution of the OUTL BUS, A instruction.
$\overline{PSEN}$ (Program Store Enable)	Output	A signal to fetch an instruction from external program memory is output. (Active "0" level)
$\overline{RESET}$	Output	$\overline{RESET}$ input initialize the processor. (Active "0" level) Used to terminate the power-down mode.
$\overline{SS}$ (Single Step)	Output	A program is executed step by step. This pin can also be used to control internal oscillation when the power-down mode is reset. (Active "0" level)
EA (External Access)	Input	When held at high level, all instructions are fetched from external memory. (Active "1" level)
$\overline{PROG}$ (Expander Strobe)	Output	This output strobes the I/O expander MSM82C43RS.

Designation	Input/Output	Function
XTAL 1 (Crystal 1)	Input	One side of crystal input for internal oscillator. External source can also be input.
XTAL 2 (Crystal 2)	Input	Other side of Crystal input for internal oscillator.
V_{CC}	—	Power supply terminal
V_{DD}	—	Standby control input. Normally, "1" level. As this pin is set to "0" level, oscillation is stopped and processor goes into standby mode.
V_{SS}	—	GND

Note: The required $\overline{\text{RESET}}$ pulse duration is at least two machine cycles under the condition that power supply and oscillation are stable.

ADDED FUNCTIONS OF MSM80C48RS AND MSM80C49RS

The MSM80C48RS and MSM80C49RS basically incorporates the capabilities of Intel's 8048 and 8049, plus the following new functions:

1. Power-Down Mode Enhancements

1.1 Power-down by software

(1) Clock (See item 4, "Power-down mode", for details.)
   a. Crystal-controlled oscillator halt (HLTS instruction)
      Power requirements can be minimized.
   b. Clock supply halt (HALT instruction)
      Restart is accomplished without oscillator wait.

(2) I/O ports (See Table 4-1 and 4-2 for details.)
   I/O port floating instructions
   Power consumption resulting from inputs/outputs can be minimized with FLT and FLTT instructions.
   Port floating is cancelled by executing FRES instruction, "0" level at $\overline{INT}$ pin or "0" level at $\overline{RESET}$ pin.

(3) Six types of power-down can be done by a combination of HLTS/HALT and FLT/FLTT instructions.

1.2 Power-down by hardware (See 4.3, Power-down mode by V_{DD} pin utilization for details.)

Crystal-controlled oscillators can be halted by controlling the V_{DD} terminal, thereby floating all I/O ports for minimum power consumption.

2. Additional Instructions (11)

HLTS	MOV A, P2
HALT	MOV P1, @ R3
FLT	MOVP1 P, @ R3
FLTT	DEC @ Rr
FRES	DJNZ @ R, addr
MOV A, P1	

3. Improved Uses of BUS $P_{0\sim7}$, $P1_{0\sim7}$, $P2_{0\sim7}$, and SS terminals

3.1 BUS $P_{0\sim7}$

The MSM80C48RS and MSM80C49RS remove the limitation on the use of OUTL BUS, A instructions during the external ROM access mode by having an independent data latch and external ROM mode address latch in BUS $P_{0\sim7}$.

Consequently, there is no need to relocate bus port instructions when in the external ROM access mode.

3.2 $P1_{0\sim7}$ and $P2_{0\sim7}$

The MSM80C48RS and MSM80C49RS are designed to minimize power consumption when $P1_{0\sim7}$ and $P2_{0\sim7}$ are used as input/output ports, to maximize the performance of CMOS.

When these ports are used as output ports, the acceleration circuit is actuated only when output data turns from "0" to "1", thus speeding up the rise-up of output signals.

When these ports are used as input ports, the internal pullup resistance becomes approximately 9 kΩ when input data is "1".

The internal pullup resistance rises to approximately 100 kΩ when input data is "0". Like this, a high noise margin can be obtained by selecting the impedance and thus the outflow of current is minimized whenever these ports are used as output or input ports.

3.3 Clock generation control via the $\overline{SS}$ terminal

When crystal-controlled oscillator is halted in the HLTS or hardware power-down mode, the $\overline{SS}$ terminal is pulled down by resistor of 20 – 50 kΩ, while its internal pullup resistor of 200 – 500 kΩ is isolated from V_{CC}. When the power-down mode is cancelled, the internal resistor of $\overline{SS}$ terminal is changed from pull-down to pullup. Consequently, the CPU can be held halted for any period of time until crystal-controlled oscillator resume normal oscillation when a capacitor is connected to the $\overline{SS}$ terminal.

4. Power-Down Mode

The MSM80C48RS and MSM80C49RS power down mode can be enabled in 2 different ways-through software by a combination of clock control and port floating instructions, and through hardware by control of the V_{DD} pin.

4.1 Software power down mode

Power down mode can be done by a combination of the following instructions.

(1) HALT (clock supply halt to control circuit)

Instruction code: | 0 0 0 0 0 0 0 1 |

Description: Although crystal-controlled oscillator operation is continued, the clock supply to the CPU control circuit is halted and CPU operations suspended. When cancelling this software mode, restart is accomplished without oscillator wait. Timing charts are outlined in Figs. 4-1 and 4-2.

(2) HLTS (oscillation stop)

Instruction code: | 1 0 0 0 0 0 1 0 |

Description: The oscillator operation is halted and CPU operations suspended. In cancelling this power down mode, connecting a capacitor to the $\overline{SS}$ pin enables a reasonable wait period to be accompolished before normal operation is resumed. [Except

the case of using $\overline{\text{RESET}}$ pin] Timing charts are outlined in Figs. 4-3 and 4-4.

(3) FLT (floating of $P1_{0\sim7}$, $P2_{0\sim7}$, and $BP_{0\sim7}$)

Instruction code: 1 0 1 0 0 0 1 0

Description:

	Internal ROM mode	External ROM mode
P1	Floating	Floating
P2	Floating	$P2_{0\sim3}$ operation $P2_{4\sim7}$ floating
BP	Floating	Operation

Details of IC pin status as a result of executing the FLT instruction are shown in Table 4-1.

(4) FLTT (floating of all output pins)

Instruction code: 1 1 0 0 0 0 1 0

Description:

	Internal ROM mode	External ROM mode
ALE	Floating	Operation
$\overline{\text{PSEN}}$	Floating	Operation
PROG	Floating	Floating
$\overline{\text{WR}}$	Floating	Floating
$\overline{\text{RD}}$	Floating	Floating
TO OUT	Floating	Floating
P1	Floating	Floating
P2	Floating	$P2_{0\sim3}$ operation $P2_{4\sim7}$ floating
BP	Floating	Operation
XTAL	Operation	Operation

Details of IC pin status as a result of executing the FLTT instruction are shown in Table 4-2.

Example 1: Power down mode accomplished by stopping oscillation.
- ○ Setting by execution of HLTS-[82H] instruction.

Example 2: Power down mode accomplished by stopping the clock supply to the CPU control circuit.
- ○ Setting by execution of HALT-[01H] instruction.

Example 3: Power down mode by floating of $P1_{0\sim7}$, $P2_{0\sim7}$ and $BP_{0\sim7}$, and subsequent stopping of CPU oscillation.
- ○ Setting by first executing the FLT[A2H] instruction and then the HLTS[82H] instruction.

Example 4: Power down mode by flowing $P1_{0\sim7}$, $P2_{0\sim7}$ and $BP_{0\sim7}$, and then stopping the clock supply to the CPU control circuit.
- ○ Setting by first executing the FLT[A2H] instruction, and then the HALT[01H] instruction.

Example 5: Power down mode by floating all output pins, followed by stopping oscillation.
- ○ Setting by first executing the FLTT[C2H] instruction followed by execution of the HLTS[82H] instruction.

Example 6: Power down mode by floating all output pins, followed by stopping of the clock supply to the CPU control circuit.
- ○ Setting by first executing the FLTT[C2H] instruction, followed by execution of the HALT[01H] instruction.

4.2 Cancellation of software power down mode

The power down mode status (outlined above in examples 1 to 6) can be cancelled by using either the interrupt pin or the $\overline{\text{RESET}}$ pin.

(1) Use of the $\overline{\text{INT}}$ pin during external interrupt enabled mode (i.e. following execution of ENI instruction).
- ○ The clock generator is activated and the CPU started up when a "0" level is applied to the $\overline{\text{INT}}$ pin. If this "0" level is maintained until at least 2 ALE output signals occur, an external interrupt is generated, and execution proceeds from address 3. If, however, the power down mode has been done during the interrupt processing routine, execution is resumed just after the power down instruction.

(2) Use of the $\overline{\text{INT}}$ pin during external interrupt disabled mode (i.e. following execution of DIS I instruction or hardware reset)
- ○ The clock generator is activated and the CPU started up when a "0" level is applied to the $\overline{\text{INT}}$ pin. If this "0" level is maintained until at least 2 ALE output signals occur, execution is resumed just after the powerdown instruction.

(3) Use of the $\overline{\text{RESET}}$ pin
- ○ The clock generator is activated and the CPU started up when a "0" level is applied to the $\overline{\text{RESET}}$ pin. If this "0" level is maintained until at least 2 ALE output signals occur, the CPU is reset and execution exceeds from address 0. In case cancellation is done in oscillation stop mode, the "0" level must be input to the $\overline{\text{RESET}}$ pin till oscillation is stabilized.

1P	T0	Active	Active
2P	XTAL 1	Active	Active
3P	XTAL 2	Active	Active
4P	$\overline{\text{RESET}}$	Active	Active
5P	$\overline{\text{SS}}$	200 ~ 500 kΩ pullup	200 ~ 500 kΩ pullup
6P	$\overline{\text{INT}}$	Active	Active
7P	EA	Active	Active
8P	$\overline{\text{RD}}$	Active	Active
9P	$\overline{\text{PSEN}}$	Active	Active
10P	$\overline{\text{WR}}$	Active	Active
11P	ALE	Active	Active
12P	DB0	Floating	Active
13P	DB1	Floating	Active
14P	DB2	Floating	Active
15P	DB3	Floating	Active
16P	DB4	Floating	Active
17P	DB5	Floating	Active
18P	DB6	Floating	Active
19P	DB7	Floating	Active
20P	V_{SS}	0 [V]	0 [V]
21P	P20	Floating	Active
22P	P21	Floating	Active
23P	P22	Floating	Active
24P	P23	Floating	Active
25P	PROG	Active	Active
26P	V_{DD}	"1" level	"1" level
27P	P10	Floating	Floating
28P	P11	Floating	Floating
29P	P12	Floating	Floating
30P	P13	Floating	Floating
31P	P14	Floating	Floating
32P	P15	Floating	Floating
33P	P16	Floating	Floating
34P	P17	Floating	Floating
35P	P24	Floating	Floating
36P	P25	Floating	Floating
37P	P26	Floating	Floating
38P	P27	Floating	Floating
39P	T1	Active	Active
40P	V_{CC}	+3 to +6 [V]	+3 to +6 [V]

Note: The FLT mode itself is reset by executing the FRES instruction, or supplying "0" level to $\overline{\text{INT}}$ or $\overline{\text{RESET}}$ pin.

Table 4-2 Details of Pin Status Following Execution of FLTT Instruction

Pin No.	Pin Name	Internal ROM	External ROM
1P	T0	Floating if output enabled	Floating if output enabled
2P	XTAL 1	Active	Active
3P	XTAL 2	Active	Active
4P	$\overline{\text{RESET}}$	200 to 500 kΩ pullup	200 to 500 kΩ pullup
5P	SS		
6P	$\overline{\text{INT}}$	Active	Active
7P	EA	Active	Active
8P	$\overline{\text{RD}}$	Floating	Floating
9P	$\overline{\text{PSEN}}$	Floating	Active
10P	$\overline{\text{WR}}$	Floating	Floating
11P	ALE	Floating	Active
12P	DB 0	Floating	Active
13P	DB 1	Floating	Active
14P	DB 2	Floating	Active
15P	DB 3	Floating	Active
16P	DB 4	Floating	Active
17P	DB 5	Floating	Active
18P	DB 6	Floating	Active
19P	DB 7	Floating	Active
20P	V_{SS}	0 [V]	0 [V]
21P	P20	Floating	Active
22P	P21	Floating	Active
23P	P22	Floating	Active
24P	P23	Floating	Active
25P	PROG	Floating	Floating
26P	V_{DD}	"1" level	"1" level
27P	P10	Floating	Floating
28P	P11	Floating	Floating
29P	P12	Floating	Floating
30P	P13	Floating	Floating
31P	P14	Floating	Floating
32P	P15	Floating	Floating
33P	P16	Floating	Floating
34P	P17	Floating	Floating
35P	P24	Floating	Floating
36P	P25	Floating	Floating
37P	P26	Floating	Floating
38P	P27	Floating	Floating
39P	T1	Active	Active
40P	V_{CC}	+3 to +6 [V]	+3 to +6 [V]

Note: The FLTT mode itself is reset by executing the FRES instruction, or supplying "0" level to $\overline{\text{INT}}$ or $\overline{\text{RESET}}$ pin.

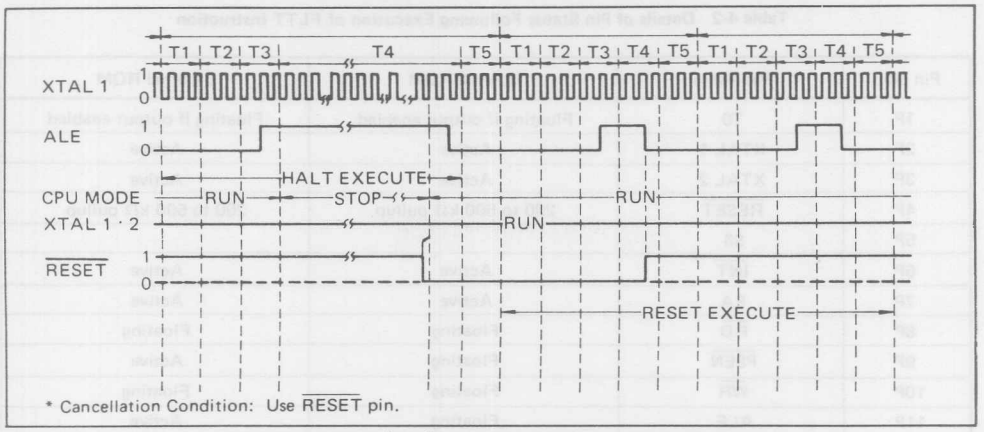


Fig. 4-1 HALT [01H] Instruction Execution Timing Chart

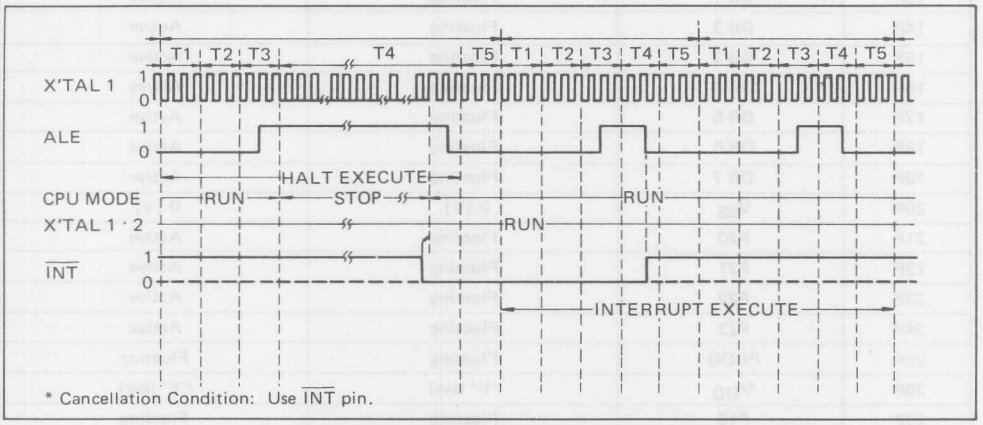


Fig. 4-2 HALT [01H] Instruction Execution Timing Chart

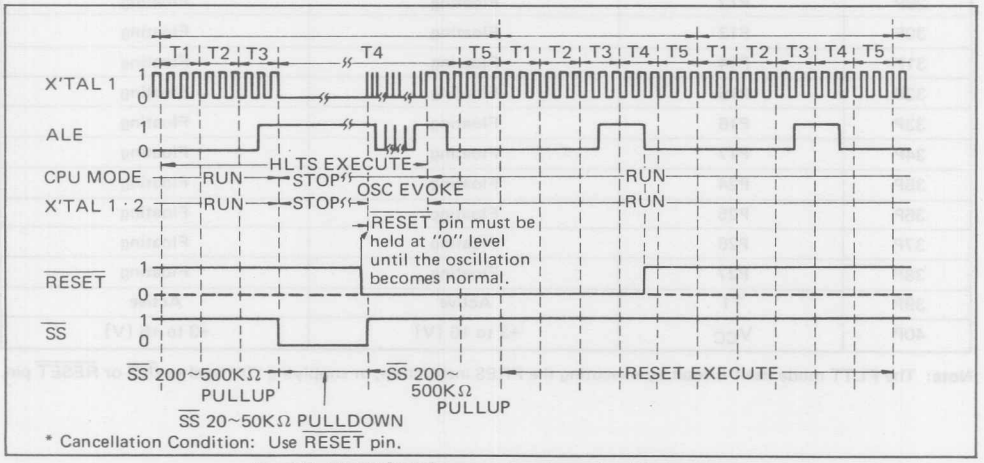


Fig. 4-3 HLTS [82H] Instruction Execution Timing Chart

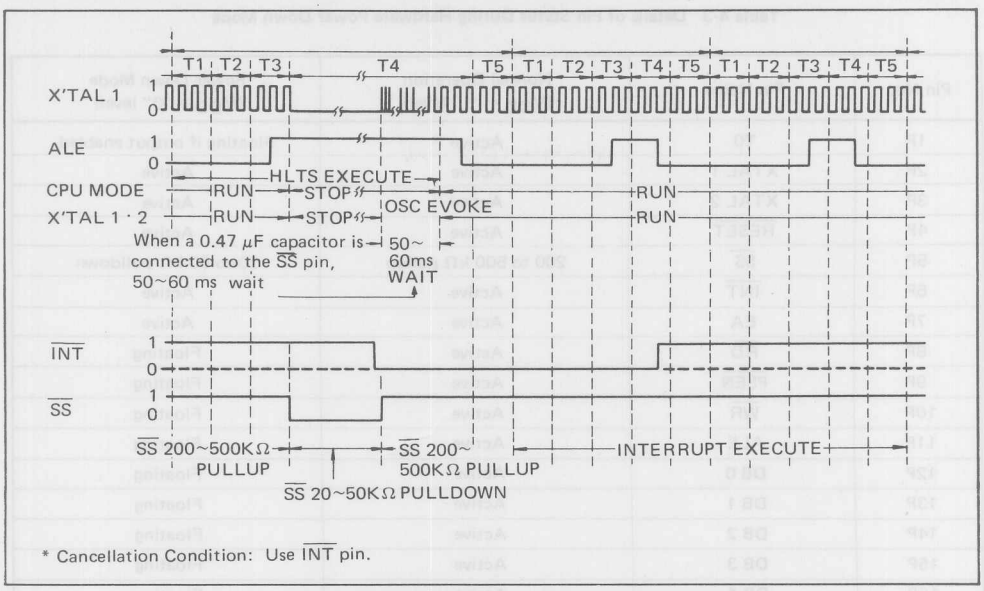


Fig. 4-4 HLTS [82H] Instruction Execution Timing Chart

4.3 Hardware power down mode

In the MSM80C48RS and MSM80C49RS, forcing the level at the V_{DD} pin [pin 26] to a "0" during either external ROM or internal ROM mode results in suspension of the oscillator function and subsequent floating (high impedance) of all the I/O pins except the $\overline{RESET}$, $\overline{SS}$ and XTAL 1/2 pins. The CPU is thereby stopped while maintaining internal status. Details of the IC pin status at this time are outlined in Table 4-3.

4.4 Cancellation of hardware power down mode

(1) Use of $\overline{RESET}$ pin

- The clock generator is activated and the CPU started up when a "1" level is applied to the V_{DD} pin while a "0" level is input to the $\overline{RESET}$ pin. If this "0" level is kept applied to the $\overline{RESET}$ pin until oscillation become stable, the CPU will be reset and will start executing from address 0. The timing chart is outlined in Fig. 4-5.

(2) Use of the $\overline{INT}$ pin during external interrupt enabled status (i.e. following execution of ENI instruction)

- The clock generator is activated and the CPU started up when a "1" level is applied to the V_{DD} pin while a "0" level is applied to the $\overline{INT}$ pin. If this "0" level is maintained until at least 2 ALE output signals occur, an external interrupt is generated, and execution starts from address 3. However, if the power down mode is started during an interrupt processing routine, execution will be continued on the next instruction after the present instruction. The timing chart is outlined in Fig. 4-6.

(3) Use of the $\overline{INT}$ pin during external interrupt disabled mode (i.e. following execution of DIS I instruction or hardware reset)

- The clock generator is activated and the CPU started up when a "1" level is applied to the V_{DD} pin while a "0" level is applied to the $\overline{INT}$ pin. If this "0" level is maintained until at least 2 ALE output signals occur, execution is continued on the next instruction after the present instruction. The timing chart is outlined in Fig. 4-6.

(4) Use of V_{DD} pin only

- The clock generator is activated and the CPU started up when a "1" level is applied to the V_{DD} pin while a "1" level is also applied to both the $\overline{RESET}$ and $\overline{INT}$ pins. In this case, execution is resumed on the stopped position. The timing chart is outlined in Fig. 4-7.

Pin No.	Pin Name	Normal Operation (V_{DD} = "1" level)	Power Down Mode (V_{DD} = "0" level)
1P	T0	Active	Floating if output enabled
2P	XTAL 1	Active	Active
3P	XTAL 2	Active	Active
4P	$\overline{\text{RESET}}$	Active	Active
5P	$\overline{\text{SS}}$	200 to 500 kΩ pullup	20 to 50 kΩ pulldown
6P	$\overline{\text{INT}}$	Active	Active
7P	EA	Active	Active
8P	$\overline{\text{RD}}$	Active	Floating
9P	$\overline{\text{PSEN}}$	Active	Floating
10P	$\overline{\text{WR}}$	Active	Floating
11P	ALE	Active	Floating
12P	DB 0	Active	Floating
13P	DB 1	Active	Floating
14P	DB 2	Active	Floating
15P	DB 3	Active	Floating
16P	DB 4	Active	Floating
17P	DB 5	Active	Floating
18P	DB 6	Active	Floating
19P	DB 7	Active	Floating
20P	V_{SS}	0 [V]	0 [V]
21P	P20	Active	Floating
22P	P21	Active	Floating
23P	P22	Active	Floating
24P	P23	Active	Floating
25P	PROG	Active	Floating
26P	V_{DD}	"1" level	"0" level
27P	P10	Active	Floating
28P	P11	Active	Floating
29P	P12	Active	Floating
30P	P13	Active	Floating
31P	P14	Active	Floating
32P	P15	Active	Floating
33P	P16	Active	Floating
34P	P17	Active	Floating
35P	P24	Active	Floating
36P	P25	Active	Floating
37P	P26	Active	Floating
38P	P27	Active	Floating
39P	T1	Active	Active
40P	V_{CC}	+3 to +6 [V]	+3 to +6 [V]

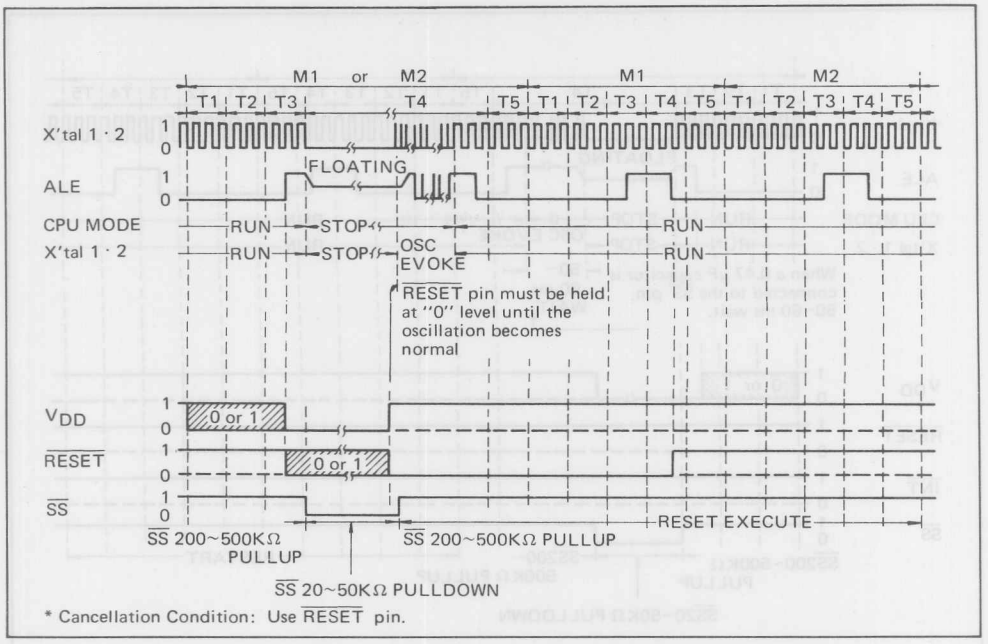


Fig. 4-5 Hardware Power Down Mode Timing Chart

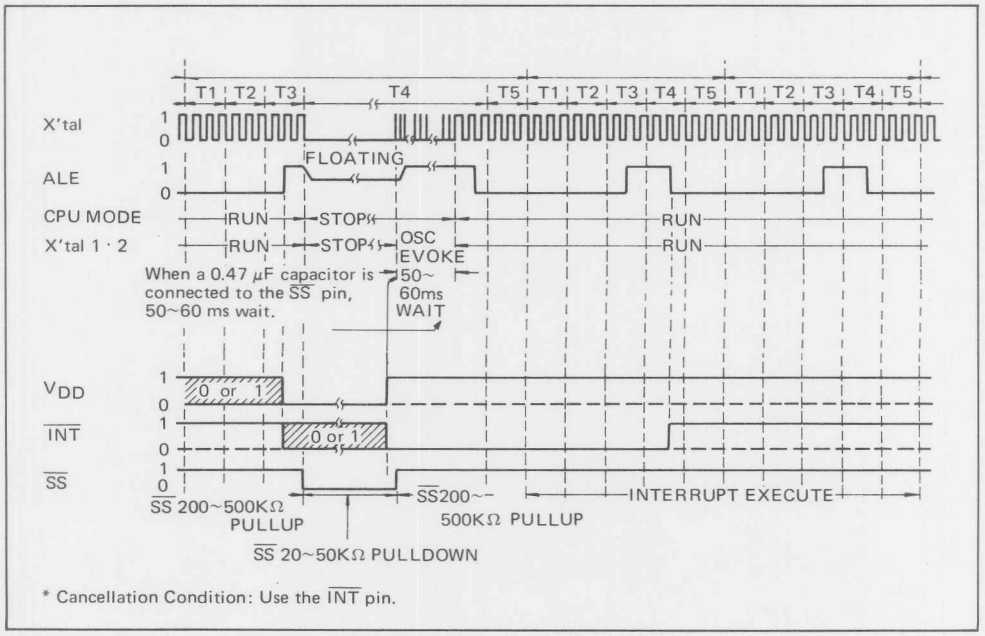


Fig. 4-6 Hardware Power Down Mode Timing Chart

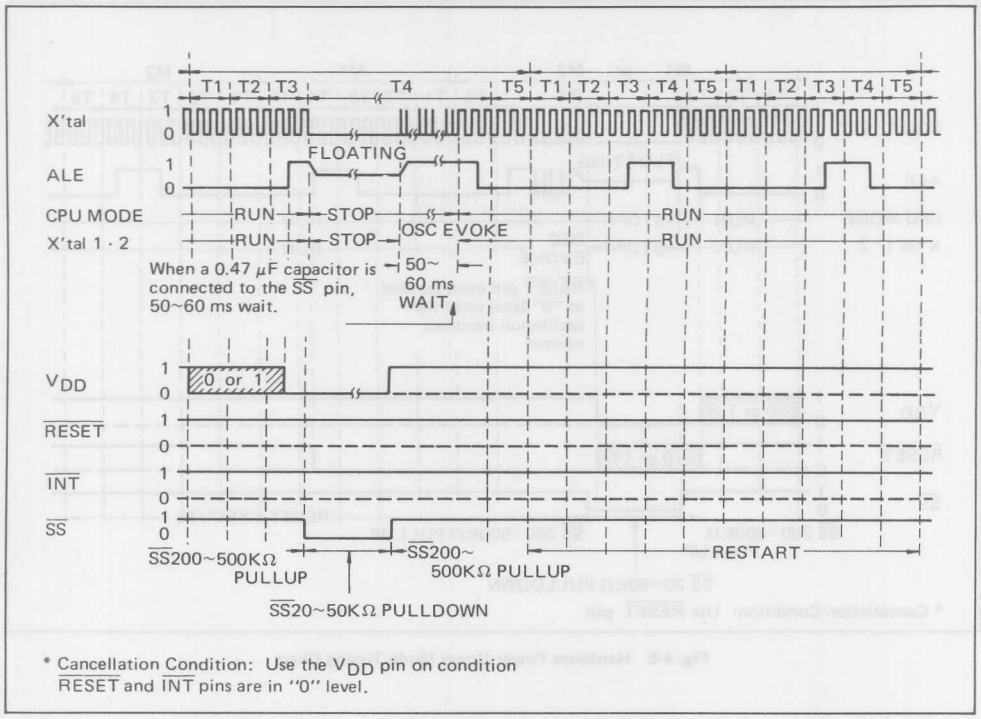


Fig. 4-7 Hardware Power Down Mode Timing Chart

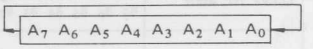
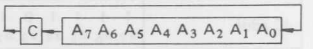
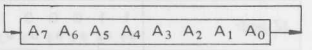
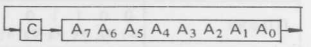
MSM80C48RS/MSM80C49RS INSTRUCTION TABLE

H \ L	0 0 0 0 0	1 0 0 0 1	2 0 0 1 0	3 0 0 1 1	4 0 1 0 0	5 0 1 0 1	6 0 1 1 0	7 0 1 1 1	8 1 0 0 0	9 1 0 0 1	A 1 0 1 0	B 1 0 1 1	C 1 1 0 0	D 1 1 0 1	E 1 1 1 0	F 1 1 1 1
0 0000	NOP	HALT Added	OUTL BUS, A	ADD A, # data	JMP	EN I		DEC A	INS A, BUS	IN A, P1	IN A, P2		MOVD A, PP			
1 0001	INC @, R0	INC @, R1	JB0 addr	ADDC A, # data	CAL	DIS I	JTF addr	INC A	INC R							
2 0010	XCHA@,R0	XCHA@,R1		MOV A, # data	JMP	EN TCNTI	JNTO addr	CLR A	XCH A, R							
3 0011	XCHD A, @R0	XCHD A, @ R1	JB1 addr		CAL	DIS TCNTI	JTO addr	CPL A		OUTL P1, A	OUTL P2, A		MOVD PP, A			
4 0100	ORL A, @R0	ORL A, @R1	MOV A, T	ORL A, # data	JMP	STRT CNT	JNT1 addr	SWP A	ORL A, R							
5 0101	ANL A, @R0	ANL A, @ R1	JB2 addr	ANL A, # data	CAL	STRT T	JT1 addr	DA A	ANL A, R							
6 0110	ADD A, @ R0	ADD A, @ R1	MOV T, A	MOV A, P1 Added	JMP	STOP TCNT		RRC A	ADD A, Rr							
7 0111	ADDC A, @ R0	ADDC A, @ R1	JB3 addr	MOV A, P2 Added	CAL	ENTO CLK	JF1 addr	RR A	ADDC A, R							
8 1000	MOVX A, @ R0	MOVX A, @ R1	HLTS Added	RET	JMP	CLR F	JNI addr		ORL BUS, # data	ORL P1, # data	ORL P2, # data		ORLD PP, A			
9 1001	MOVX @ R0, A	MOVX @ R1, A	JB4 addr	RETR	CAL	CPL F0	JNZ addr	CLR C	ANL BUS, # data	ANL P1, # data	ANL P2, # data		ANL D PP, A			
A 1010	MOV @ R0, A	MOV @R1, A	FLT Added	MOVP A @A	JMP	CLR F1		CPL C	MOV R, A							
B 1011	MOV @ R0, # data	MOV @R1, # data	JB5 addr	JMPP @A	CAL	CPL F1	JF0 addr		MOV R, # data							
C 1100	DEC @R0 Added	DEC @R1 Added	FLTT Added	MOVP1 P@ R3 Added	JMP	SEL RB0	JZ addr	MOV A, PSW	DEC R							
D 1101	XRL A, @ R0	XRL A, @ R1	JB6 addr	XRLA, # data	CAL	SEL RB1		MOV PSW, A	XRL A, R							
E 1110	DJNZ @R0 Added	DJNZ @R1 Added	FRES Added	MOVP3 A, @A	JMP	SEL MB0	JNC addr	RL A	DJNZ R							
F 1111	MOV A, @ R0	MOV A, @ R1	JB7 addr	MOVP1, @R3 Added	CAL	SEL MB1	JC addr	RLC A	MOV A, R							

A	: Accmulator	PC	: Program counter
AC	: Auxiliary carry	Pp	: Port indicator (p = 4 ~ 7)
addr	: 12-bit program memory address or its part	PSW	: Program status word
Bb	: Bit indicator (b = 0 ~ 7)	Rr	: Resister indicator (r = 0 ~ 7)
Bs	: Bank switch	SP	: Stack pointer
BUS	: BUS PORT	T	: Timer
C	: Carry	TF	: Timer flag
CLK	: Clock	T0, T1	: Test pins T0 and T1
CNT	: Counter	X	: External RAM
D	: 4-bit data	#	: Symbol denoting immediate data
data	: 8-bit numerical value	@	: Symbol denoting indirect address
DBF	: Memory data bank flip-flop	(X)	: Denotes contents of X
F0, F1	: F0 flag and F1 flag	((X))	: Address contents addressed by X
I	: Interrupt	←	: Transference

LIST OF INSTRUCTIONS

Classification	Mnemonic	Instruction Code $D_7\ D_6\ D_5\ D_4\ D_3\ D_2\ D_1\ D_0$	Hexadecimal	Byte	Cycle	Description	Page
Accumulator operation instructions	ADD A, Rr	0 1 1 0 1 r_2 r_1 r_0	68 ~ 6F	1	1	(AC), (C), (A) ← (A) + (Rr)	
	ADD A, @Rr	0 1 1 0 0 0 0 r_0	60 ~ 61	1	1	(AC), (C), (A) ← (A) + ((Rr))	
	ADD A, # data	0 0 0 0 0 0 1 1 d_7 d_6 d_5 d_4 d_3 d_2 d_1 d_0	03 Byte 2	2	2	(AC), (C), (A) ← (A) + data	
	ADDC A, Rf	0 1 1 1 1 r_2 r_1 r_0	78 ~ 7F	1	1	(AC), (C), (A) ← (A) + (Rr) + (C)	
	ADDC A, @Rr	0 1 1 1 0 0 0 r_0	70 ~ 71	1	1	(AC), (C), (A) ← (A) + ((Rr)) + (C)	
	ADDC A, # data	0 0 0 1 0 0 1 1 d_7 d_6 d_5 d_4 d_3 d_2 d_1 d_0	13 Byte 2	2	2	(AC), (C), (A) ← (A) + data + (C)	
	ANL A, Rr	0 1 0 1 1 r_2 r_1 r_0	58 ~ 5F	1	1	(A) ← (A) AND (Rr)	
	ANL A, @Rr	0 1 0 1 0 0 0 r_0	50 ~ 51	1	1	(A) ← (A) AND ((Rr))	
	ANL A, # data	0 1 0 1 0 0 1 1 d_7 d_6 d_5 d_4 d_3 d_2 d_1 d_0	53 Byte 2	2	2	(A) ← (A) AND data	
	ORL A, Rr	0 1 0 0 1 r_2 r_1 r_0	48 ~ 4F	1	1	(A) ← (A) OR (Rr)	
	ORL A, @Rr	0 1 0 0 0 0 0 r_0	40 ~ 41	1	1	(A) ← (A) OR ((Rr))	
	ORL A, # data	0 1 0 0 0 0 1 1 d_7 d_6 d_5 d_4 d_3 d_2 d_1 d_0	43 Byte 2	2	2	(A) ← (A) OR data	
	XRL A, Rr	1 1 0 1 1 r_2 r_1 r_0	D8 ~ DF	1	1	(A) ← (A) XOR (Rr)	
	XRL A, @Rr	1 1 0 1 0 0 0 r_0	D0 ~ D1	1	1	(A) ← (A) XOR ((Rr))	
	XRL A, # data	1 1 0 1 0 0 1 1 d_7 d_6 d_5 d_4 d_3 d_2 d_1 d_0	D3 Byte 2	2	2	(A) ← (A) XOR data	
	INC A	0 0 0 1 0 1 1 1	17	1	1	(A) ← (A) + 1	
	DEC A	0 0 0 0 0 1 1 1	07	1	1	(A) ← (A) – 1	
	CLR A	0 0 1 0 0 1 1 1	27	1	1	(A) ← 0	
	CPL A	0 0 1 1 0 1 1 1	37	1	1	(A) ← $(\overline{A})$	
	DA A	0 1 0 1 0 1 1 1	57	1	1	Add 6 to bits 0 ~ 3 when contents of accumulator bits 0 ~ 3 exceed 9 or when auxiliary carry (AC) is 1. Then add 6 to bits 4 ~ 7 when the result of adding the carry from the lower 0 ~ 3 exceeds 9, or when carry (C) is 1. Set 1 in the carry flag if an overflow is generated in the end result, or when the carry prior to adjustment is 1.	

Classification	Mnemonic	Instruction Code D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0	Hexadecimal	Byte	Cycle	Description	Page
Accumulator operation instructions	SWAP A	0 1 0 0 0 1 1 1	47	1	1	$(A_{4\sim7}) \leftrightarrows (A_{0\sim3})$	
	RL A	1 1 1 0 0 1 1 1	E7	1	1	 Rotate accumulator contents to the left by 1 bit.	
	RLC A	1 1 1 1 0 1 1 1	F7	1	1	 Rotate accumulator with carry contents to the left by 1 bit.	
	RR A	0 1 1 1 0 1 1 1	77	1	1	 Rotate accumulator contents to the right by 1 bit.	
	RRC A	0 1 1 0 0 1 1 1	67	1	1	 Rotate accumulator with carry contents to the right by 1 bit.	
Input/output instructions	IN A, P1	0 0 0 0 1 0 0 1	09	1	2	(A) ← (P1)	
	IN A, P2	0 0 0 0 1 0 1 0	0A	1	2	(A) ← (P2)	
	OUTL P1, A	0 0 1 1 1 0 0 1	39	1	2	(P1) ← (A)	
	OUTL P2, A	0 0 1 1 1 0 1 0	3A	1	2	(P2) ← (A)	
	ANL P1, # data	1 0 0 1 1 0 0 1 d_7 d_6 d_5 d_4 d_3 d_2 d_1 d_0	99 Byte 2	2	2	(P1) ← (P1) AND data	
	ANL P2, # data	1 0 0 1 1 0 1 0 d_7 d_6 d_5 d_4 d_3 d_2 d_1 d_0	9A Byte 2	2	2	(P2) ← (P2) AND data	
	ORL P1, # data	1 0 0 0 1 0 0 1 d_7 d_6 d_5 d_4 d_3 d_2 d_1 d_0	89 Byte 2	2	2	(P1) ← (P1) OR data	
	ORL P2, # data	1 0 0 0 1 0 1 0 d_7 d_6 d_5 d_4 d_3 d_2 d_1 d_0	8A Byte 2	2	2	(P2) ← (P2) OR data	
	INS A, BUS	0 0 0 0 1 0 0 0	08	1	2	(A) ← (BUS)	
	OUTL BUS, A	0 0 0 0 0 0 1 0	02	1	2	(BUS) ← (A)	
	ANL BUS, # data	1 0 0 1 1 0 0 0 d_7 d_6 d_5 d_4 d_3 d_2 d_1 d_0	98 Byte 2	2	2	(BUS) ← (BUS) AND data	
	ORL BUS, # data	1 0 0 0 1 0 0 0 d_7 d_6 d_5 d_4 d_3 d_2 d_1 d_0	88 Byte 2	2	2	(BUS) ← (BUS) OR data	
	MOVD A, Pp	0 0 0 0 1 1 P_1 P_0	0C~0F	1	2	$(A_{0\sim3}) \leftarrow (Pp)$ p=4~7 $(A_{4\sim7}) \leftarrow 0$	
	MOVD Pp, A	0 0 1 1 1 1 P_1 P_0	3C~3F	1	2	$(Pp) \leftarrow (A_{0\sim3})$ p = 4 ~ 7	
	ANLD Pp, A	1 0 0 1 1 1 P_1 P_0	9C~9F	1	2	$(Pp) \leftarrow (Pp)$ AND $(A_{0\sim3})$ p = 4 ~ 7	
	ORLD Pp, A	1 0 0 0 1 1 P_1 P_0	8C~8F	1	2	$(Pp) \leftarrow (Pp)$ OR $(A_{0\sim3})$ p = 4 ~ 7	
Register operation instructions	INC Rr	0 0 0 1 1 r_2 r_1 r_0	18~1F	1	1	(Rr) ← (Rr) + 1	
	INC @Rr	0 0 0 1 0 0 0 r_0	10~11	1	1	((Rr)) ← ((Rr)) + 1	
	DEC Rr	1 1 0 0 1 r_2 r_1 r_0	C8~CF	1	1	(Rr) ← (Rr) – 1	
	DEC @Rr	1 1 0 0 0 0 0 r_0	C0~C1	1	1	((Rr)) ← ((Rr)) – 1	
Branching instructions	JMP addr	a_{10} a_9 a_8 0 0 1 0 0 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0		2	2	$(PC_{8\sim10})$ ← addr 8 ~ 10 $(PC_{0\sim7})$ ← addr 0 ~ 7 (PC_{11}) ← DBF	
	JMPP @A	1 0 1 1 0 0 1 1	B3	1	2	$(PC_{0\sim7})$ ← ((A))	

Classification	Mnemonic	Instruction Code D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0	Hexadecimal	Byte	Cycle	Description	Page
Branching instructions	DJNZ Rr, addr	1 1 1 0 1 r_2 r_1 r_0 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0	E8~EF Byte 2	2	2	(Rr) ←(Rr) – 1 ($PC_{0\sim7}$) ←addr if (Rr) ≠ 0 (PC) ←(PC) + 2 if (Rr) = 0	
	DJNZ @Rr, addr	1 1 1 0 0 0 0 r_0 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0	E0~E1 Byte 2	2	2	((Rr)) ←((Rr)) – 1 ($PC_{0\sim7}$) ←addr if ((Rr)) ≠ 0 (PC) ←(PC) + 2 if ((Rr)) = 0	
	JC addr	1 1 1 1 0 1 1 0 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0	F6 Byte 2	2	2	($PC_{0\sim7}$) ←addr if C = 1 (PC) ←(PC)+2 if C = 0	
	JNC addr	1 1 1 0 0 1 1 0 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0	E6 Byte 2	2	2	($PC_{0\sim7}$) ←addr if C = 0 (PC) ←(PC)+2 if C = 1	
	JZ addr	1 1 0 0 0 1 1 0 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0	C6 Byte 2	2	2	($PC_{0\sim7}$) ←addr if A = 0 (PC) ←(PC)+2 if A = 0	
	JNZ addr	1 0 0 1 0 1 1 0 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0	96 Byte 2	2	2	($PC_{0\sim7}$) ←addr if A = 0 (PC) ←(PC)+2 if A = 0	
	JT0 addr	0 0 1 1 0 1 1 0 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0	36 Byte 2	2	2	($PC_{0\sim7}$) ←addr if T0 = 1 (PC) ←(PC)+2 if T0 = 0	
	JNT0 addr	0 0 1 0 0 1 1 0 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0	26 Byte 2	2	2	($PC_{0\sim7}$) ←addr if T0 = 0 (PC) ←(PC)+2 if T0 = 1	
	JT1 addr	0 1 0 1 0 1 1 0 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0	56 Byte 2	2	2	($PC_{0\sim7}$) ←addr if T1 = 1 (PC) ←(PC)+2 if T1 = 0	
	JNT1 addr	0 1 0 0 0 1 1 0 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0	46 Byte 2	2	2	($PC_{0\sim7}$) ←addr if T1 = 0 (PC) ←(PC)+2 if T1 = 1	
	JF0 addr	1 0 1 1 0 1 1 0 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0	B6 Byte 2	2	2	($PC_{0\sim7}$) ←addr if F0 = 1 (PC) ←(PC)+2 if F0 = 0	
	FJ1 addr	0 1 1 1 0 1 1 0 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0	76 Byte 2	2	2	($PC_{0\sim7}$) ←addr if F1 = 1 (PC) ←(PC)+2 if F1 = 0	
	JTF addr	0 0 0 1 0 1 1 0 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0	16 Byte 2	2	2	($PC_{0\sim7}$) ←addr TF ←0 } if TF = 1 (PC) ←(PC)+2 if TF = 0	
	JN1 addr	1 0 0 0 0 1 1 0 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0	86 Byte 2	2	2	($PC_{0\sim7}$) ←addr if $\overline{INT}$ = 0 (PC) ←(PC)+2 if $\overline{INT}$ = 1	
	JBb addr	b_2 b_1 b_0 1 0 0 1 0 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0		2	2	($PC_{0\sim7}$) ←addr if Bb = 1 (PC) ←(PC)+2 if Bb = 0	
Sub-routine instructions	CALL addr	a_{10} a_9 a_8 1 0 1 0 0 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0		2	2	((SP)) ←(PC), ($PSW_{4\sim7}$) ($PC_{8\sim10}$) ←addr 8 ~ 10 ($PC_{0\sim7}$) ←addr 0 ~ 7 (PC_{11}) ←DBF (SP) ←(SP) + 1	
	RET	1 0 0 0 0 0 1 1	83	1	2	(SP) ←(SP) – 1 (PC) ←((SP))	
	RETR	1 0 0 1 0 0 1 1	93	1	2	(SP) ←(SP) – 1 (PC) ←((SP)) ($PSW_{4\sim7}$) ←((SP)) INT END	
Flag operation instructions	CLR C	1 0 0 1 0 1 1 1	97	1	1	(C) ←0	
	CPL C	1 0 1 0 0 1 1 1	A7	1	1	(C) ←($\overline{C}$)	
	CLR F0	1 0 0 0 0 1 0 1	85	1	1	(F0) ←0	
	CPL F0	1 0 0 1 0 1 0 1	95	1	1	(F0) ←($\overline{F0}$)	
	CLR F1	1 0 1 0 0 1 0 1	A5	1	1	(F1) ←0	
	CPL F1	1 0 1 1 0 1 0 1	B5	1	1	(F1) ←($\overline{F1}$)	
Data transfer instructions	MOV A, Rr	1 1 1 1 1 r_2 r_1 r_0	F8~FF	1	1	(A) ←(Rr)	
	MOV A, @Rr	1 1 1 1 0 0 0 r_0	F0~F1	1	1	(A) ←((Rr))	
	MOV A, # data	0 0 1 0 0 0 1 1 d_7 d_6 d_5 d_4 d_3 d_2 d_1 d_0	23 Byte 2	2	2	(A) ←data	

Classification	Mnemonic	Instruction Code D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0	Hexadecimal	Byte	Cycle	Description	Page
Data transfer instructions	MOV Rr, A	1 0 1 0 1 r_2 r_1 r_0	A8~AF	1	1	(Rr) ←(A)	
	MOV @Rr, A	1 0 1 0 0 0 0 r_0	A0~A1	1	1	((Rr)) ←(A)	
	MOV Rr, # data	1 0 1 1 1 r_2 r_1 r_0 d_7 d_6 d_5 d_4 d_3 d_2 d_1 d_0	B8~BF Byte 2	2	2	(Rr) ←data	
	MOV @Rr, # data	1 0 1 1 0 0 0 r_0 d_7 d_6 d_5 d_4 d_3 d_2 d_1 d_0	B0~B1 Byte 2	2	2	((Rr)) ←data	
	MOV A, PSW	1 1 0 0 0 1 1 1	C7	1	1	(A) ←(PSW)	
	MOV PSW, A	1 1 0 1 0 1 1 1	D7	1	1	(PSW) ←(A)	
	XCH A, Rr	0 0 1 0 1 r_2 r_1 r_0	28~2F	1	1	(A) ⇆(Rr)	
	XCH A, @Rr	0 0 1 0 0 0 0 r_0	20~21	1	1	(A) ⇆((Rr))	
	XCHD A, @Rr	0 0 1 1 0 0 0 r_0	30~31	1	1	($A_{0\sim3}$) ⇆(($Rr_{0\sim3}$))	
	MOVX A, @Rr	1 0 0 0 0 0 0 r_0	80~81	1	2	(A) ← ((Rr)) External RAM	
	MOVX @Rr, A	1 0 0 1 0 0 0 r_0	90~91	1	2	((Rr)) ← (A) External RAM	
	MOVP A, @A	1 0 1 0 0 0 1 1	A3	1	2	(A) ←((PC$_{8\sim10}$, A))	
	MOVP3 A, @A	1 1 1 0 0 0 1 1	E3	1	2	(A) ←((PC$_{011}$, A))	
	MOVP1 P, @R3	1 1 0 0 0 0 1 1	C3	1	2	(P1) ←(((PC$_{0\sim7}$) ←((R3))))	
	MOV P1, @R3	1 1 1 1 0 0 1 1	F3	1	2	(P1) ←((R3))	
	MOV A, P1	0 1 1 0 0 0 1 1	63	1	1	(A) ←(P1) Latch data	
	MOV A, P2	0 1 1 1 0 0 1 1	73	1	1	(A) ←(P2) Latch data	
Control instructions	EN TCNTI	0 0 1 0 0 1 0 1	25	1	1	TINT Enable F/F ←1	
	DIS TCNTI	0 0 1 1 0 1 0 1	35	1	1	TINT Enable F/F ←0	
	ENI	0 0 0 0 0 1 0 1	05	1	1	EXINT Enable F/F ←1	
	DIS 1	0 0 0 1 0 1 0 1	15	1	1	EXINT Enable F/F ←0	
	SEL RB0	1 1 0 0 0 1 0 1	C5	1	1	(BS) ←0	
	SEL RB1	1 1 0 1 0 1 0 1	D5	1	1	(BS) ←1	
	SEL MB0	1 1 1 0 0 1 0 1	E5	1	1	(DBF) ←0	
	SEL MB1	1 1 1 1 0 1 0 1	F5	1	1	(DBF) ←1	
	ENTO CLK	0 1 1 1 0 1 0 1	75	1	1	TO ←1/3 XTAL 1	
	FLT	1 0 1 0 0 0 1 0	A2	1	1	P1, P2, BUS Floating	
	FLTT	1 1 0 0 0 0 1 0	C2	1	1	CPU Out Signal Floating	
	FRES	1 1 1 0 0 0 1 0	E2	1	1	FLT, FLTT RESET	
	HALT	0 0 0 0 0 0 0 1	01	1	1	CPU Control Clock Stop	
	HLTS	1 0 0 0 0 0 1 0	82	1	1	XTAL 1 · 2 Stop	
Time/counter instructions	MOV A, T	0 1 0 0 0 0 1 0	42	1	1	(A) ←(T)	
	MOV T, A	0 1 1 0 0 0 1 0	62	1	1	(T) ←(A)	
	STRT T	0 1 0 1 0 1 0 1	55	1	1	(T) ← [÷32] ← [÷15] ←XTAL	
	STRT CNT	0 1 0 0 0 1 0 1	45	1	1	(T) ←T1 Clock in Count	
	STOP TCNT	0 1 1 0 0 1 0 1	65	1	1	(T) Count Stop	
Other instruction	NOP	0 0 0 0 0 0 0 0	00	1	1	(PC) ←(PC) + 1	

Supply Voltage	V_{CC}	Ta = 25°C	−0.3 to 7	V
Input Voltage	V_I	Ta = 25°C	−0.3 to V_{CC}	V
Storage Temperature	T_{stg}		−55 to +150	°C

OPERATING RANGE

Parameter	Symbol	Conditions	Limits	Unit
Supply Voltage	V_{CC}		+2.5 to +6	V
RAM Retention Voltage	V_{CC}		+2 to +6	V
Ambient Temperature	T_A		−40 to +85	°C
Fan Out	N	MOS load	10	
		TTL load	1	

DC ELECTRICAL CHARACTERISTICS

(V_{CC} = 5V ±10%, Ta = –40 to +85°C)

Parameter	Symbol	Conditions	Min.	Typ.	Max.	Unit	Measurement circuit
"L" Input Voltage	V_{IL}		–0.3		0.8	V	
"H" Input Voltage (1)	V_{IH}		2.2		V_{CC}	V	
"H" Input Voltage (2)	V_{IH}		3.8		V_{CC}	V	
"L" Output Voltage (3)	V_{OL}	I_{OL} = 2 mA			0.45	V	
"L" Output Voltage (4)	V_{OL}	I_{OL} = 1.6 mA			0.45	V	1
"H" Output Voltage (3)	V_{OH}	I_{OH} = 400 μA	2.4			V	
"H" Output Voltage (4)	V_{OH}	I_{OH} = 50 μA	2.4			V	
"H" Output Voltage (3)	V_{OH}	I_{OH} = 20 μA	4.2			V	
"H" Output Voltage (4)	V_{OH}	I_{OH} = 10 μA	4.2			V	
Input Leak Current	I_{IL}	$V_{SS} \leqq V_{IN} \leqq V_{CC}$			±10	μA	2
Output Leak Current (5)	I_{OL}	$V_{SS} \leqq V_O \leqq V_{CC}$			±10	μA	3
$\overline{\text{RESET}}$ Pull up Resistance	R_R	$V_{IN} \geqq V_{IH}$/ $V_{IN} \leqq V_{IL}$	20/500		50/750	kΩ	2
$\overline{\text{SS}}$ Pull up Resistance (6)	R_{SS}	Oscillation stop/oscillation	20/200		50/500	kΩ	
P1, P2 Pull up Resistance	$R_{P1, P2}$	$V_{IN} \geqq V_{IH}$/ $V_{IN} \leqq V_{IL}$	5/75		15/150	kΩ	3
Power Supply Current	I_{CC}	At hardware power down V_{CC} = 2V (TA = +25°C)		1	10	μA	4
		At MLTS execution *7 V_{CC}=2V (TA = +25°C)		1	10	μA	
		At HALT (6 MHz)		1.5	3	mA	
		At HALT (11 MHz)		2.5	5	mA	
		At execution (6 MHz)		5	10	mA	
		At execution (11 MHz)		10	20	mA	

Notes: (1) This does not apply to $\overline{\text{RESET}}$, XTAL1, XTAL2, and V_{DD}.
(2) $\overline{\text{RESET}}$, XTAL1, XTAL2, V_{DD}
(3) BUS, $\overline{\text{RD}}$, $\overline{\text{WR}}$, $\overline{\text{PSEN}}$, ALE
(4) Other outputs
(5) High-impedance state
(6) This operates as a pull-down resistor when the oscillation is stopped the oscillation is stopped in the HLTS or hardware power-down mode and as a pull-up resistor in other states.
(7) This does not contain flow out current from I/O Ports and Signal pins.

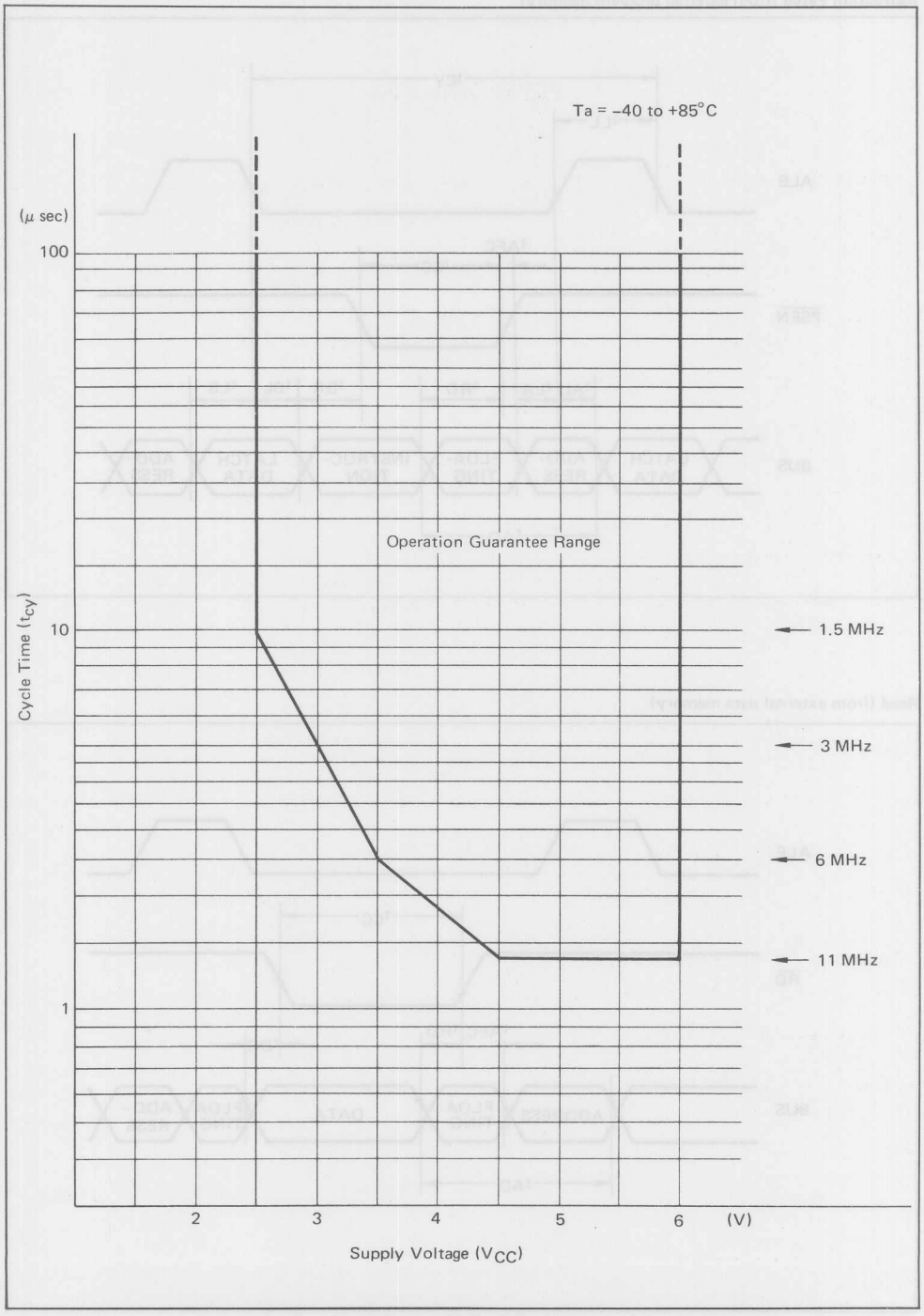
AC CHARACTERISTICS

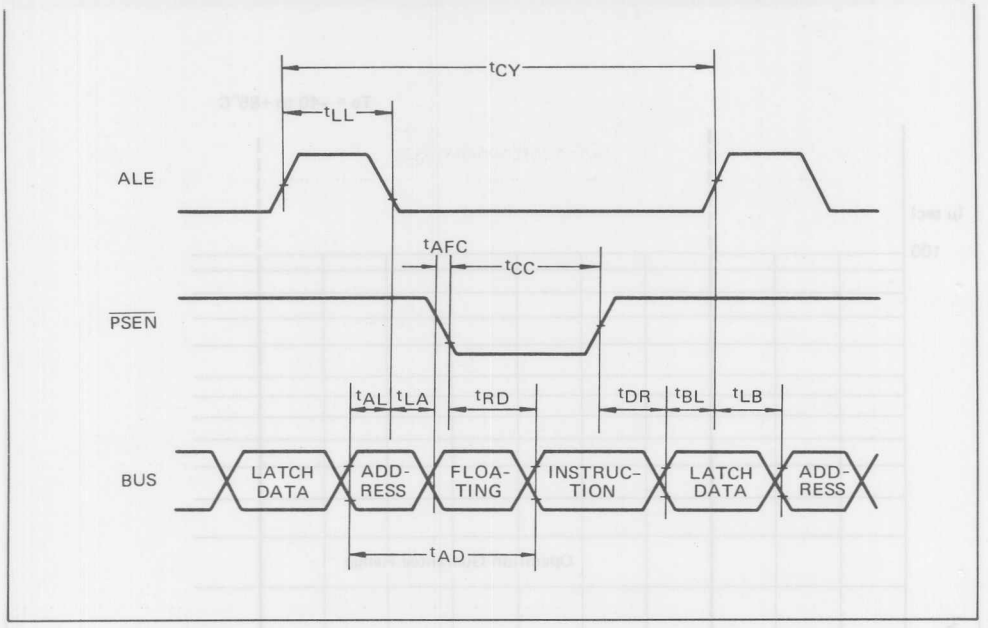
(V_{CC} = 5V ±10%, T_A = −40°C to +85°C)

Parameter	Symbol	Limits				Unit
		11 MHz Clock		Variable Clock (0–11 MHz)		
		Min.	Max.	Min.	Max.	
Cycle Time	t_{CY}	1.36		1.36		μs
ALE Pulse Width	t_{LL}	150		$7/30t_{CY}-165$		ns
Address Setup to ALE	t_{AL}	70		$2/15t_{CY}-110$		ns
Address Hold from ALE	t_{LA}	50		$1/15t_{CY}-40$		ns
Bus Port Latch Data Setup to ALE	t_{BL}	110		$5/30t_{CY}-115$		ns
Bus Port Latch Data Hold from ALE	t_{LB}	90		$3/30t_{CY}-45$		ns
Control Pulse Width ($\overline{PSEN}$, $\overline{RD}$, and $\overline{WR}$)	t_{CC}	300		$6/15t_{CY}-245$		ns
Data Setup before $\overline{WR}$	t_{DW}	250		$6/15t_{CY}-295$		ns
Data Hold after $\overline{WR}$	t_{WD}	40		$2/15t_{CY}-140$		ns
Data Hold after $\overline{RD}$	t_{DR}	0	100	0	100	ns
$\overline{PSEN}$, $\overline{RD}$ to Data-in	t_{RD}		200		$5/15t_{CY}-250$	ns
Address Setup to $\overline{WR}$	t_{AW}	200		$6/15t_{CY}-345$		ns
Address Setup to Data-in	t_{AD}		400		$8/15t_{CY}-325$	ns
Address Float to $\overline{RD}$, $\overline{PSEN}$	t_{AFC}	0		0	0	ns
Port Control Setup to $\overline{PROG}$	t_{CP}	100		$2/15t_{CY}-80$		ns
Port Control Hold from $\overline{PROG}$	t_{PC}	60		$4/15t_{CY}-300$		ns
$\overline{PROG}$ to P2 Input Valid	t_{PR}	–	650		$9/15t_{CY}-165$	ns
Output Data Setup	t_{DP}	200		$6/15t_{CY}-345$		ns
Output Data Hold	t_{PD}	20		$3/15t_{CY}-250$		ns
Input Data Hold from $\overline{PROG}$	t_{PF}	0	150	0	150	ns
$\overline{PROG}$ Pulse Width	t_{PP}	700		$10/15t_{CY}-205$		ns
Port 2 I/O Setup to ALE	t_{PL}	150		$9/30t_{CY}-255$		ns
Port 2 I/O Hold from ALE	t_{LP}	20		$3/30t_{CY}-115$		ns

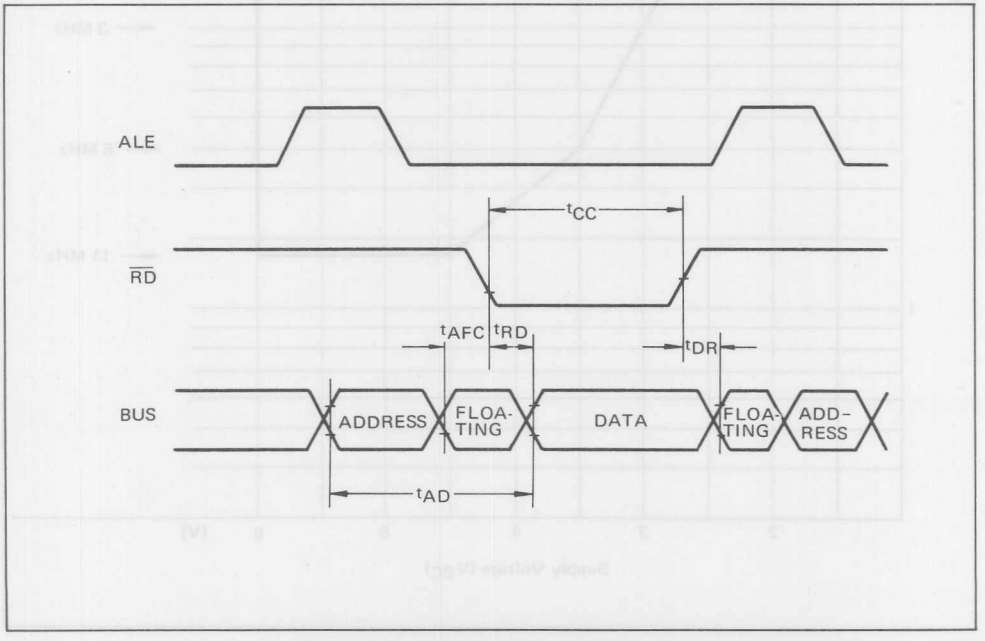
Note: Control output: C_L = 80 pF
Bus output: C_L = 150 pF [for 20 pF(t_{WD})]

MSM80C49 OPERATION GUARANTEE RANGE

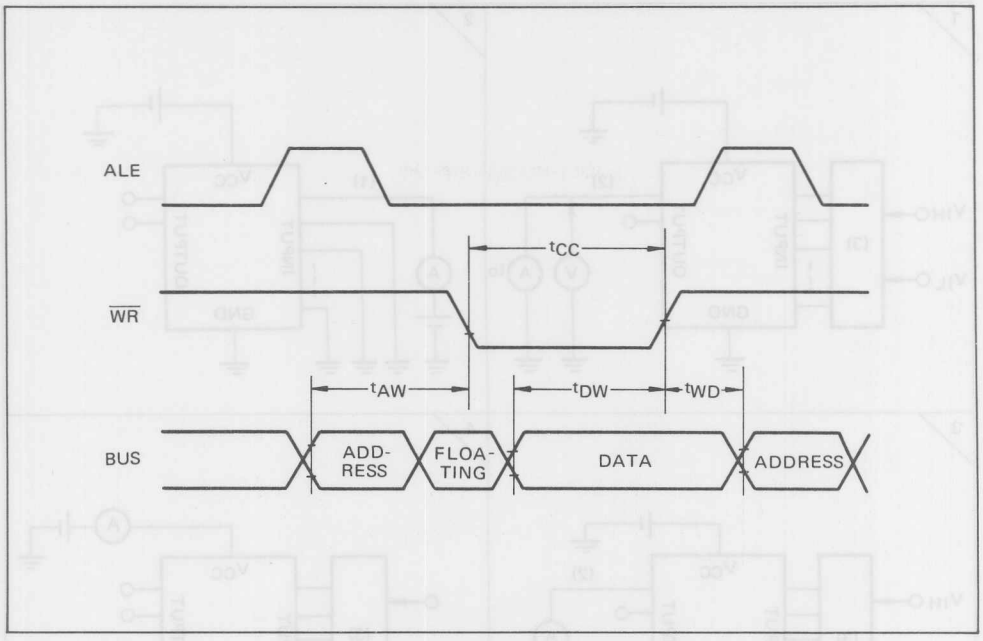




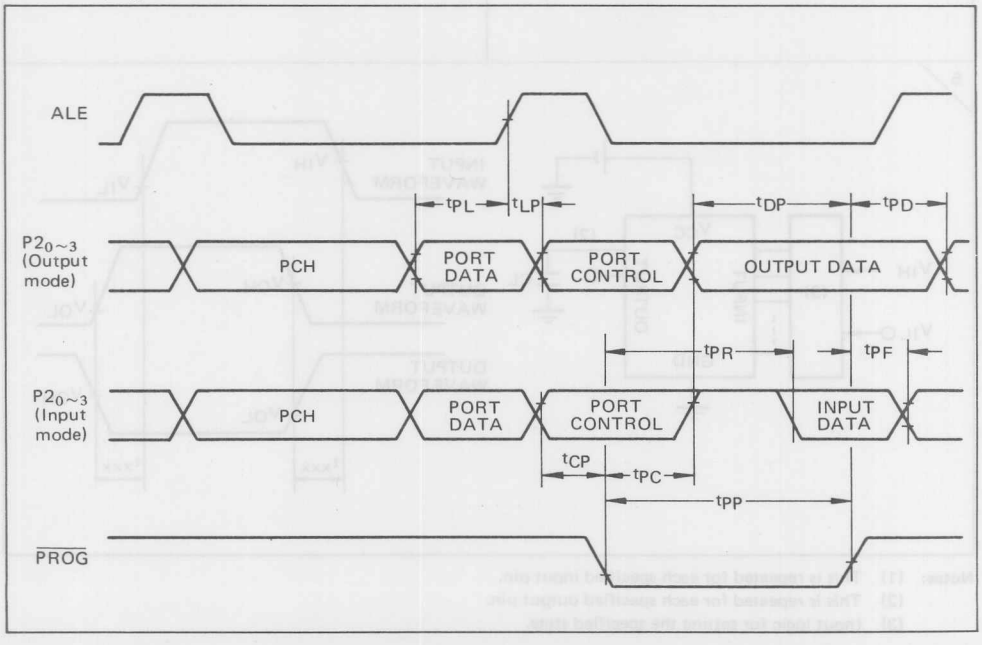
Read (from external data memory)



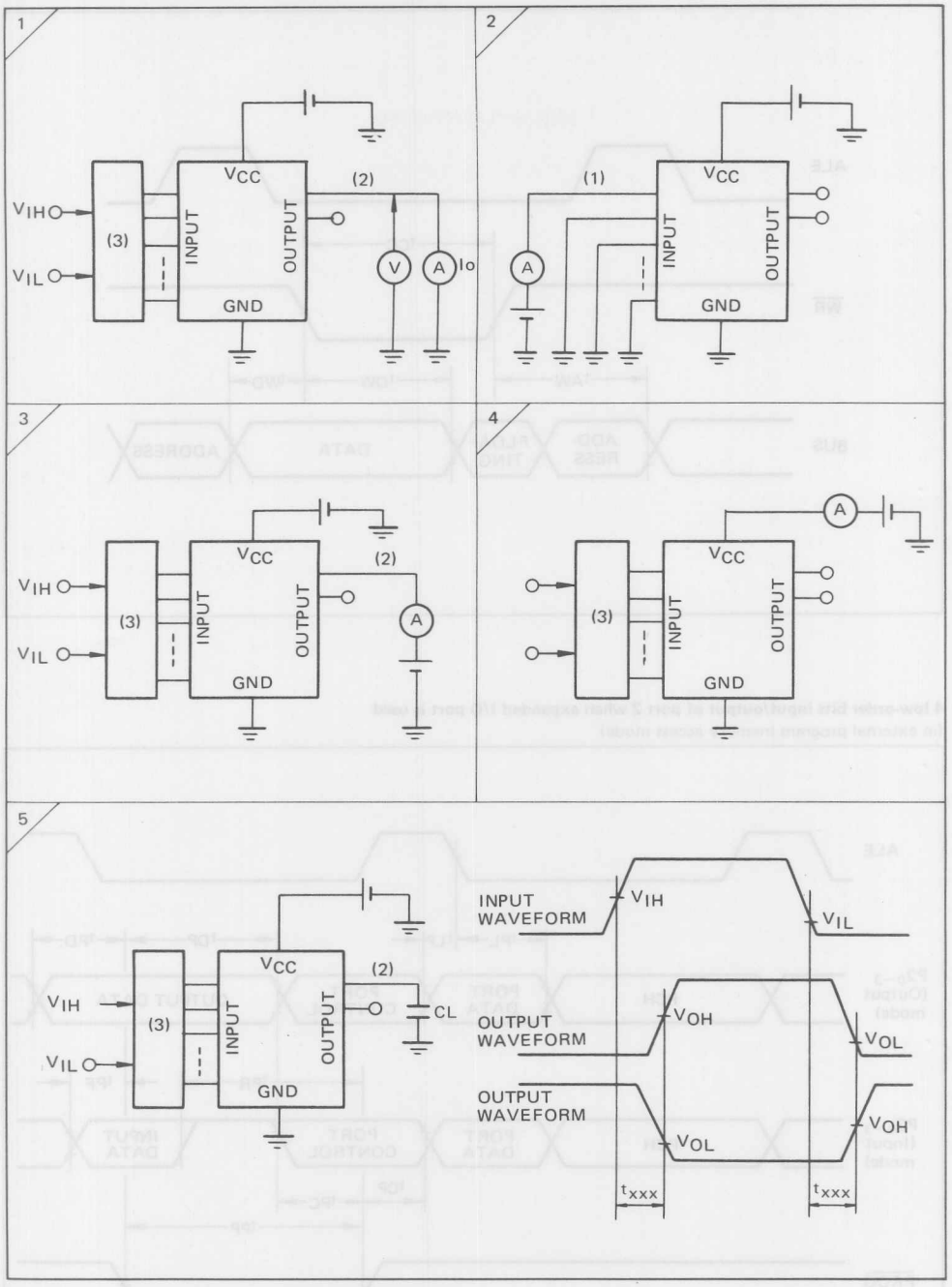
Write (to external data memory)



4 low-order bits input/output of port 2 when expanded I/O port is used
(in external program memory access mode)



MEASUREMENT CIRCUIT



Notes: (1) This is repeated for each specified input pin.
(2) This is repeated for each specified output pin.
(3) Input logic for setting the specified state.

OKI semiconductor

MSM80C85A

8-BIT CMOS MICROPROCESSOR

GENERAL DESCRIPTION

The MSM80C85A is a complete 8-bit parallel central processor implemented in silicon gate C-MOS technology and compatible with MSM8085A.

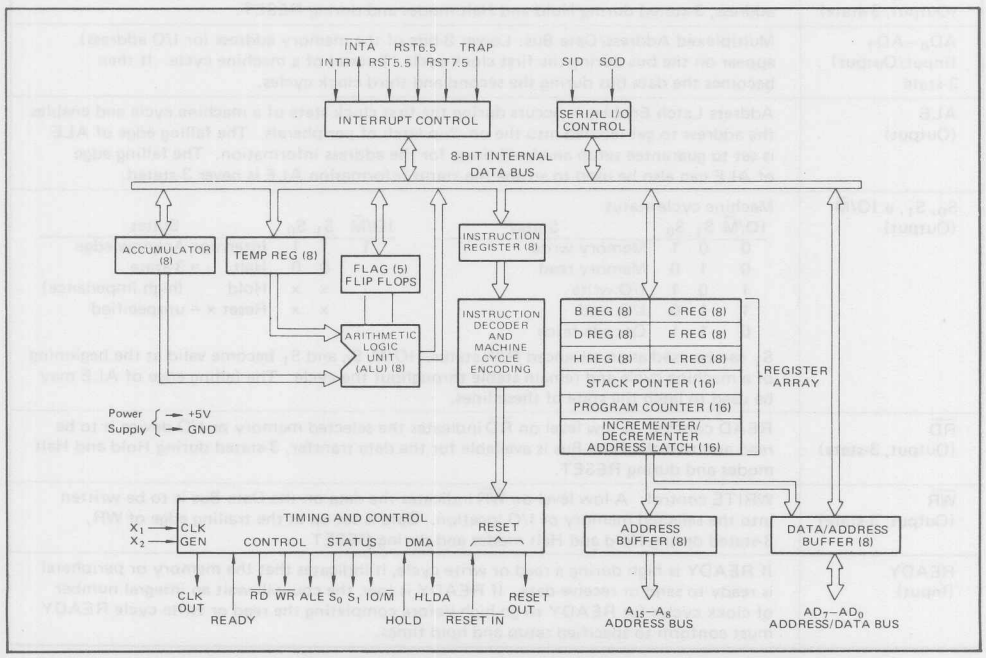
It is designed with same processing speed and lower power consumption compared with MSM8085A, thereby offering a high level of system integration.

The MSM80C85A uses a multiplexed address/data bus. The address is split between the 8-bit address bus and the 8-bit data bus. The on-chip address latches of MSM81C55/MSM81C56/MSM83C55 memory products allow a direct interface with the MSM80C85A.

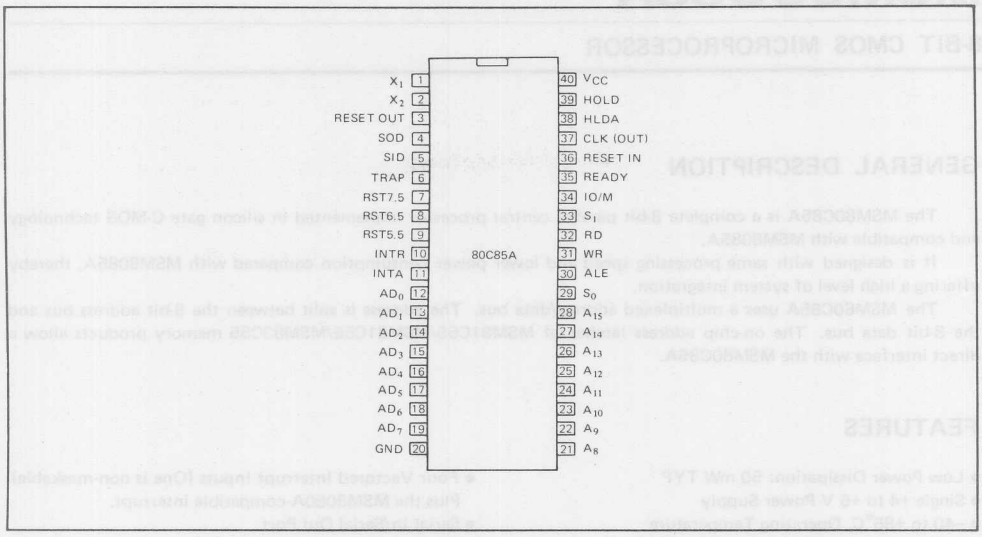
FEATURES

- Low Power Dissipation: 50 mW TYP
- Single +4 to +6 V Power Supply
- –40 to +85°C, Operating Temperature
- Compatible with MSM8085A
- 1.3μ Instruction Cycle
- On-Chip Clock Generator (with External Crystal)
- On-Chip System Controller; Advanced Cycle Status Information Available for Large System Control
- Four Vectored Interrupt Inputs (One is non-maskable) Plus the MSM8080A-compatible interrupt.
- Serial In/Serial Out Port
- Decimal, Binary and Double Precision Arithmetic
- Addressing Capability to 64K Bytes of Memory

FUNCTIONAL BLOCK DIAGRAM



PIN CONFIGURATION



MSM80C85A FUNCTIONAL PIN DEFINITION

The following describes the function of each pin:

Symbol	Function
A_8-A_{15} (Output, 3-state)	Address Bus: The most significant 8-bits of the memory address or the 8-bits of the I/O address, 3-stated during Hold and Halt modes and during RESET.
AD_0-AD_7 (Input/Output) 3-state	Multiplexed Address/Data Bus: Lower 8-bits of the memory address (or I/O address) appear on the bus during the first clock cycle (T state) of a machine cycle. It then becomes the data bus during the second and third clock cycles.
ALE (Output)	Address Latch Enable: It occurs during the first clock state of a machine cycle and enables the address to get latched into the on-chip latch of peripherals. The falling edge of ALE is set to guarantee setup and hold times for the address information. The falling edge of ALE can also be used to strobe the status information ALE is never 3-stated.
S_0, S_1, a $IO/\overline{M}$ (Output)	Machine cycle status: $IO/\overline{M}$ S_1 S_0 States 0 0 1 Memory write 0 1 0 Memory read 1 0 1 I/O write 1 1 0 I/O read 0 1 1 Opcode fetch $IO/\overline{M}$ S_1 S_0 States 1 1 1 Interrupt Acknowledge · 0 0 Halt · = 3-state · x x Hold (high impedance) · x x Reset x = unspecified S_1 can be used as an advanced R/W status. $IO/\overline{M}$, S_0 and S_1 become valid at the beginning of a machine cycle and remain stable throughout the cycle. The falling edge of ALE may be used to latch the state of these lines.
$\overline{RD}$ (Output, 3-state)	READ control: A low level on $\overline{RD}$ indicates the selected memory or I/O device is to be read and that the Data Bus is available for the data transfer, 3-stated during Hold and Halt modes and during RESET.
$\overline{WR}$ (Output, 3-state)	WRITE control: A low level on $\overline{WR}$ indicates the data on the Data Bus is to be written into the selected memory or I/O location. Data is set up at the trailing edge of WR, 3-stated during Hold and Halt modes and during RESET.
READY (Input)	If READY is high during a read or write cycle, it indicates that the memory or peripheral is ready to send or receive data. If READY is low, the cpu will wait an integral number of clock cycles for READY to go high before completing the read or write cycle READY must conform to specified setup and hold times.

Symbol	Function
HOLD (Input)	HOLD indicates that another master is requesting the use of the address and data buses. The cpu, upon receiving the hold request, will relinquish the use of the bus as soon as the completion of the current bus transfer. Internal processing can continue. The processor can regain the bus only after the HOLD is removed. When the HOLD is acknowledged, the Address, Data, $\overline{RD}$, $\overline{WR}$, and IO/$\overline{M}$ lines are 3-stated.
HLDA (Output)	HOLD ACKNOWLEDGE: Indicates that the cpu has received the HOLD request and that git will relinquish the bus in the next clock cycle. HLDA goes low after the Hold request is removed. The cpu takes the bus one half clock cycle after HLDA goes low.
INTR (Input)	INTERRUPT REQUEST: Is used as a general purpose interrupt. It is sampled only during the next to the last clock cycle of an instruction and during Hold and Halt states. If it is active, the Program Counter (PC) will be inhibited from incrementing and an $\overline{INTA}$ will be issued. During this cycle a RESTART or CALL instruction can be inserted to jump to the interrupt service routine. The INTR is enabled and disabled by software. It is disabled by Reset and immediately after an interrupt is accepted.
$\overline{INTA}$ (Output)	INTERRUPT ACKNOWLEDGE: Is used instead of (and has the same timing as) $\overline{RD}$ during the instruction cycle after an INTR is accepted.
RST 5.5 RST 6.5 RST 7.5 (Input)	RESTART INTERRUPTS: These three inputs have the same timing as INTR except they cause an internal RESTART to be automatically inserted. The priority of these interrupts is ordered as shown in Table 1. These interrupts have a higher priority than INTR. In addition, they may be individually masked out using the SIM instruction.
TRAP (Input)	Trap interrupt is a nonmaskable RESTART interrupt. It is recognized at the same timing as INTR or RST 5.5–7.5. It is unaffected by any mask or Interrupt Disable. It has the highest priority of any interrupt. (See Table 1.)
$\overline{RESET\ IN}$ (Input)	Sets the Program Counter to zero and resets the Interrupt Enable and HLDA flip-flops. The data and address buses and the control lines are 3-stated during RESET and because of the asynchronous nature of RESET, the processor's internal registers and flags may be altered by RESET with unpredictable results. $\overline{RESET\ IN}$ is a Schmitt-triggered input, allowing connection to an R-C network for power-on RESET delay. The cpu is held in the reset condition as long as $\overline{RESET\ IN}$ is applied.
RESET OUT (Output)	Indicates cpu is being reset. Can be used as a system reset. The signal is synchronized to the processor clock and lasts an integral number of clock periods.
X_1, X_2 (Input)	X_1 and X_2 are connected to a crystal to drive the internal clock generator. X_1 can also be an external clock input from a logic gate. The input frequency is divided by 2 to give the processor's internal operating frequency.
CLK (Output)	Clock Output for use as a system clock. The period of CLK is twice the X_1, X_2 input period.
SID (Input)	Serial input data line. The data on this line is loaded into accumulator bit 7 whenever a RIM instruction is executed.
SOD (Output)	Serial output data line. The output SOD is set or reset as specified by the SIM instruction.
V_{CC}	+5 volt supply.
GND	Ground Reference.

Table 1 Interrupt Priority, Restart Address, and Sensitivity

Name	Priority	Address Branched To (1) When Interrupt Occurs	Type Trigger
TRAP	1	24H	Rising edge and high level until sampled.
RST 7.5	2	3CH	Rising edge (latched).
RST 6.5	3	34H	High level until sampled.
RST 5.5	4	2CH	High level until sampled.
INTR	5	(2)	High level until sampled.

Notes: (1) The processor pushes the PC on the stack before branching to the indicated address.
(2) The address branched to depends on the instruction provided to the cpu when the interrupt is acknowledged.

FUNCTIONAL DESCRIPTION

The MSM80C85A is a complete 8-bit parallel central processor. It is designed with silicon gate C-MOS technology and requires a single +5 volt supply. Its basic clock speed is 3MHz, thus improving on the present MSM8080A's performance with higher system speed. Also it is designed to fit into a minimum system of three IC's: The cpu (MSM80C85A), a RAM/IO (MSM81C55/MSM81C56), and a ROM/IO chip (MSM83C55).

The MSM80C85A has twelve addressable 8-bit registers. Four of them can function only as two 16-bit register pairs. Six others can be used interchangeably as 8-bit registers or a 16-bit register pairs. The MSM-80C85A register set is as follows:

Mnemonic	Register	Contents
ACC or A	Accumulator	8-bits
PC	Program Counter	16-bit address
BC, DE, HL	General-Purpose Registers; data pointer (HL)	8-bit x 6 or 16-bits x 3
SP	Stack Pointer	16-bit address
Flags or F	Flag Register	5 flag (8-bit space)

The MSM80C85A uses a multiplexed Data Bus. The address is split between the higher 8-bit Address Bus and the lower 8-bit Address/Data Bus. During the first T state (clock cycle) of a machine cycle the low order address is sent out on the Address/Data Bus. These lower 8-bits may be latched externally by the Address Latch Enable signal (ALE). During the rest of the machine cycle the data bus is used for memory or I/O data.

The MSM80C85A provides $\overline{RD}$, $\overline{WR}$, S_0, S_1 and IO/$\overline{M}$ signals for bus control. An Interrupt Acknowledge signal ($\overline{INTA}$) is also provided. Hold and all Interrupts are synchronized with the processor's internal clock. The MSM80C85A also provides Serial Input Data (SID) and Serial Output Data (SOD) lines for a simple serial interface.

In addition to these features, the MSM80C85A has three maskable, vector interrupt pins and one nonmaskable TRAP interrupt.

INTERRUPT AND SERIAL I/O

The MSM80C85A has 5 interrupt inputs: INTR, RST 5.5, RST 6.5, RST 7.5, and TRAP. INTR is identical in function to the MSM8080A INT. Each of the three RESTART inputs, 5.5, 6.5, and 7.5, has a programmable mask. TRAP is also a RESTART interrupt but it is nonmaskable.

The three maskable interrupts cause the internal execution of RESTART (saving the program counter in the stack and branching to the RESTART address) if the interrupts are enabled and if the interrupt mask is not set. The nonmaskable TRAP causes the internal execution of a RESTART vector independent of the state of the interrupt enable or masks. (See Table 1.)

There are two different types of inputs in the restart interrupts. RST 5.5 and RST 6.5 are high level-sensitive like INTR (and INT on the MSM8080A) and are recognized with the same timing as INTR. RST 7.5 is rising edge-sensitive.

For RST 7.5, only a pulse is required to set an internal flip-flop which generates the internal interrupt request. The RST 7.5 request flip-flop remains set until the request is serviced. Then it is reset automatically. This flip-flop may also be reset by using the SIM instruction or by issuing a $\overline{RESET\ IN}$ to the MSM80C85A. The RST 7.5 internal flip-flop will be set by a pulse on the RST 7.5 pin even when the RST 7.5 interrupt is masked out.

The interrupts are arranged in a fixed priority that determines which interrupt is to be recognized if more than one is pending as follows: TRAP—highest priority, RST 7.5, RST 6.5, RST 5.5, INTR—lowest priority. This priority scheme does not take into account the priority of a routine that was started by a higher priority interrupt. RST 5.5 can interrupt an RST 7.5 routine if the interrupts are re-enabled before the end of the RST 7.5 routine.

The TRAP interrupt is useful for catastrophic events such as power failure or bus error. The TRAP input is recognized just as any other interrupt but has the highest priority. It is not affected by any flag or mask. The TRAP input is both edge and level sensitive. The TRAP input must go high and remain high until it is acknowledged. It will not be recognized again until it goes low, then high again. This avoids any false triggering due to noise or logic glitches. Figure 3 illustrates the TRAP interrupt request circuitry within the MSM80C85A. Note that the servicing of any interrupt (TRAP, RST 7.5, RST 6.5, RST 5.5, INTR) disables all future interrupts (except TRAPs) until an EI instruction is executed.

The TRAP interrupt is special in that it disables interrupts, but preserves the previous interrupt enable status. Performing the first RIM instruction following a TRAP interrupt allows you to determine whether interrupts were enabled or disabled prior to the TRAP. All subsequent RIM instructions provide current interrupt enable status. Performing a RIM instruction following INTR or RST 5.5–7.5 will provide current Interrupt Enable status, revealing that Interrupts are disabled.

The serial I/O system is also controlled by the RIM and SIM instructions. SID is read by RIM, and SIM sets the SOD data.

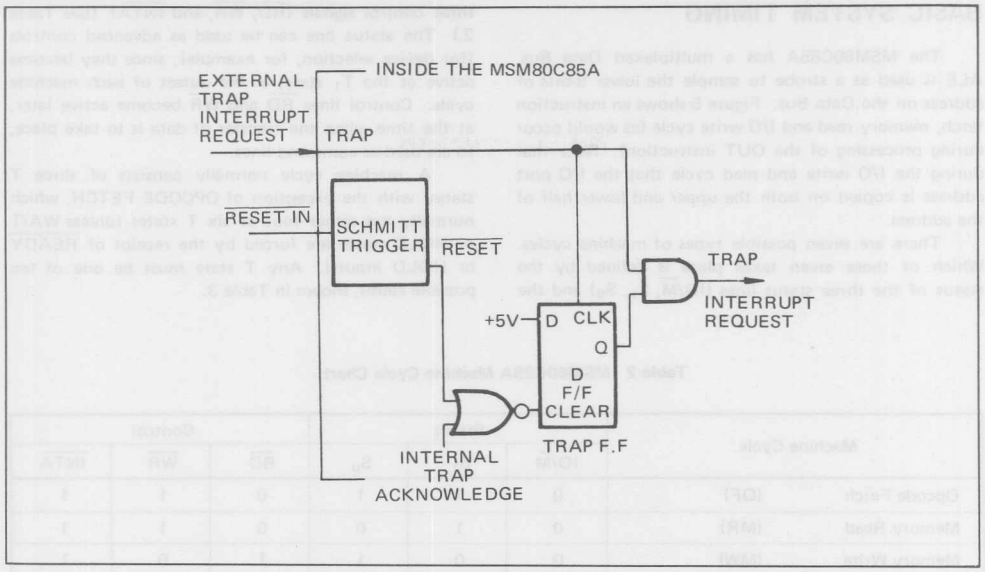


Figure 3 Trap and $\overline{\text{RESET IN}}$ Circuit

DRIVING THE X_1 and X_2 INPUTS

You may drive the clock inputs of the MSM80C-85A with a crystal, or an external clock source. The driving frequency must be at least 1 MHz, and must be twice the desired internal clock frequency; hence, the MSM80C85A is operated with a 6 MHz cyrstal (for 3 MHz clock). If a crystal is used, it must have the following characteristics:

Parallel resonance at twice the clock frequency desired
C_L (load capacitance) $\leq$ 30 pF
C_S (shunt capacitance) $\leq$ 7 pF
R_S (equivalent shunt resistance) $\leq$ 75 ohms
Drive level: 10 mW
Frequency tolerance: ±.005% (suggested)

Note the use of the capacitors between X_1, X_2 and ground. These capacitors are required to assure oscillator startup at the correct frequency.

Figure 4 shows the recommended clock driver circuits. Note in B that pullup resistor is required to assure that the high level voltage of the input is at least 4V.

For driving frequencies up to and including 6 MHz you may supply the driving signal to X_1 and leave X_2 open-circuited (Figure 4B). To prevent self-oscillation of the MSM80C85A, be sure that X_2 is not coupled back to X_1 through the driving circuit.

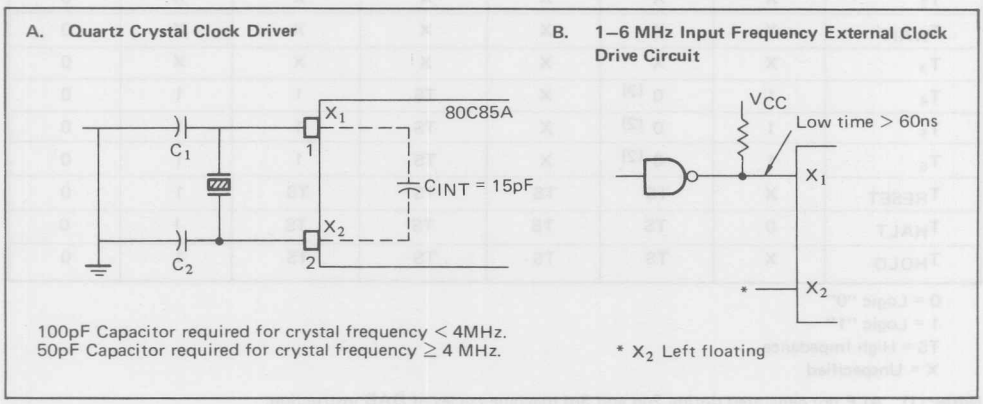


Figure 4 Clock Driver Circuits

BASIC SYSTEM TIMING

The MSM80C85A has a multiplexed Data Bus. ALE is used as a strobe to sample the lower 8-bits of address on the Data Bus. Figure 5 shows an instruction fetch, memory read and I/O write cycle (as would occur during processing of the OUT instruction). Note that during the I/O write and read cycle that the I/O port address is copied on both the upper and lower half of the address.

There are seven possible types of machine cycles. Which of these seven takes place is defined by the status of the three status lines ($IO/\overline{M}$, S_1, S_0) and the three control signals ($\overline{RD}$, $\overline{WR}$, and $\overline{INTA}$). (See Table 2.) The status line can be used as advanced controls (for device selection, for example), since they become active at the T_1 state, at the outset of each machine cycle. Control lines $\overline{RD}$ and $\overline{WR}$ become active later, at the time when the transfer of data is to take place, so are used as command lines.

A machine cycle normally consists of three T states, with the exception of OPCODE FETCH, which normally has either four or six T states (unless $\overline{WAIT}$ or HOLD states are forced by the receipt of $\overline{READY}$ or HOLD inputs). Any T state must be one of ten possible states, shown in Table 3.

Table 2 MSM80C85A Machine Cycle Chart

Machine Cycle		Status			Control		
		$IO/\overline{M}$	S_1	S_0	$\overline{RD}$	$\overline{WR}$	$\overline{INTA}$
Opcode Fetch	(OF)	0	1	1	0	1	1
Memory Read	(MR)	0	1	0	0	1	1
Memory Write	(MW)	0	0	1	1	0	1
I/O Read	(IOR)	1	1	0	0	1	1
I/O Write	(IOW)	1	0	1	1	0	1
Acknowledge of INTR	(INA)	1	1	1	1	1	0
Bus Idle	(BI): DAD	0	1	0	1	1	1
	ACK. OF RST, TRAP	1	1	1	1	1	1
	HALT	TS	0	0	TS	TS	1

Table 3 MSM80C85A Machine State Chart

Machine State	Status & Buses				Control		
	S_1, S_0	$IO/\overline{M}$	A_8–A_{15}	AD_0–AD_7	$\overline{RD}, \overline{WR}$	$\overline{INTA}$	ALE
T_1	X	X	X	X	1	1	1 (1)
T_2	X	X	X	X	X	X	0
T_{WAIT}	X	X	X	X	X	X	0
T_3	X	X	X	X	X	X	0
T_4	1	0 (2)	X	TS	1	1	0
T_5	1	0 (2)	X	TS	1	1	0
T_6	1	0 (2)	X	TS	1	1	0
T_{RESET}	X	TS	TS	TS	TS	1	0
T_{HALT}	0	TS	TS	TS	TS	1	0
T_{HOLD}	X	TS	TS	TS	TS	1	0

0 = Logic "0"
1 = Logic "1"
TS = High Impedance
X = Unspecified

Notes: (1) ALE not generated during 2nd and 3rd machine cycles of DAD instruction.
(2) $IO/\overline{M}$ = 1 during T_4~T_6 of INA machine cycle.

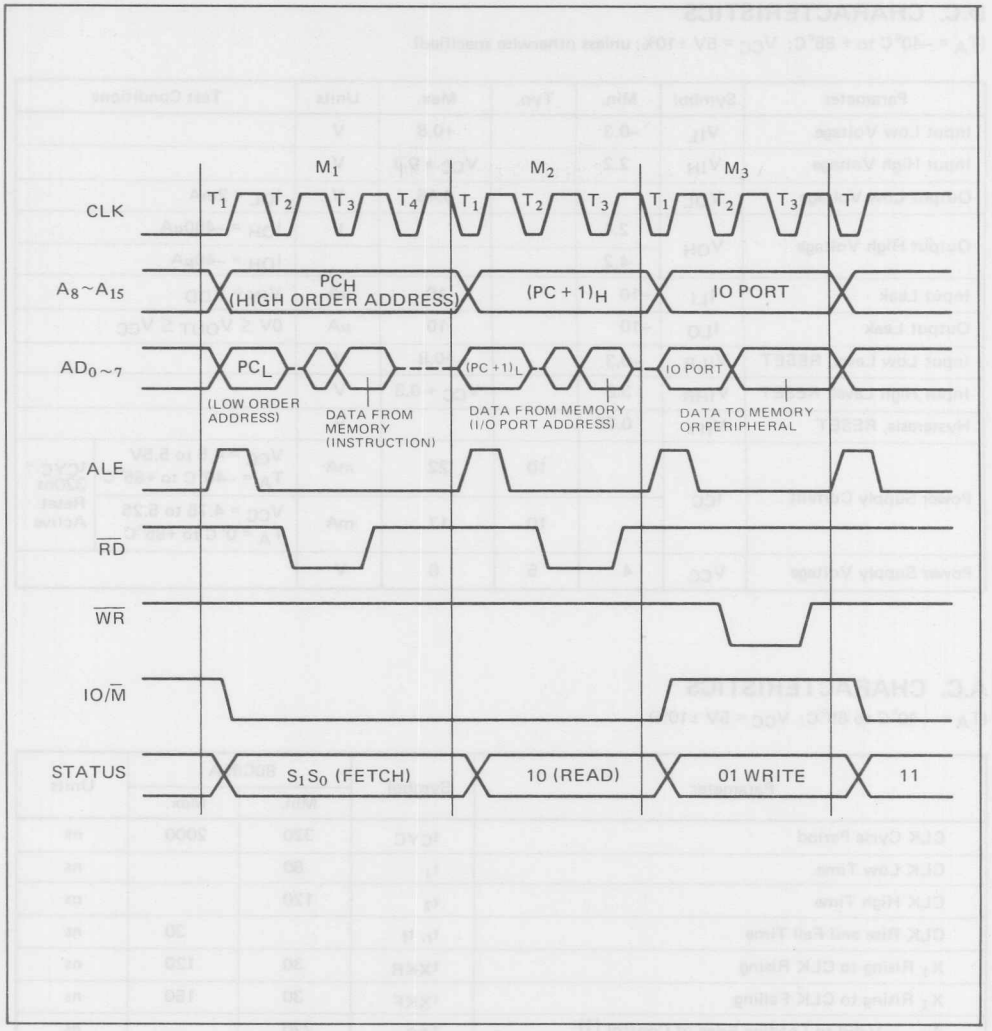


Figure 5 MSM80C85A Basic System Timing

Table 4 Absolute Maximum Ratings

Ambient Temperature Under Bias	−40°C to + 85°C
Storage Temperature	−55°C to + 150°C
Supply Voltage Respect to Ground	−0.3V to + 7.0V
Input Voltage Respect to Ground	−0.3V to V_{DD} + 0.3V
Power Dissipation	1.0 Watt

Note: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

Parameter	Symbol	Min.	Typ.	Max.	Units	Test Conditions	
Input Low Voltage	V_{IL}	–0.3		+0.8	V		
Input High Voltage	V_{IH}	2.2		V_{CC} + 0.3	V		
Output Low Voltage	V_{OL}			0.45	V	I_{OL} = 2mA	
Output High Voltage	V_{OH}	2.4			V	I_{OH} = –400μA	
		4.2				I_{OH} = –40μA	
Input Leak	I_{LI}	–10		10	μA	V_{IN} = V_{DD}	
Output Leak	I_{LO}	–10		10	μA	$0V \leq V_{OUT} \leq V_{CC}$	
Input Low Level, RESET	V_{ILR}	–0.3		+0.8	V		
Input High Level, RESET	V_{IHR}	3.0		V_{CC} + 0.3	V		
Hysteresis, RESET	V_{HY}	0.25			V		
Power Supply Current	I_{CC}		10	22	mA	V_{CC} = 4.5 to 5.5V T_A = –40°C to +85°C	t_{CYC} = 320ns Reset Active
			10	17	mA	V_{CC} = 4.75 to 5.25 T_A = 0°C to +85°C	
Power Supply Voltage	V_{CC}	4	5	6	V		

A.C. CHARACTERISTICS

(T_A = –40°C to 85°C; V_{CC} = 5V ±10%)

Parameter	Symbol	80C85A		Units
		Min.	Max.	
CLK Cycle Period	t_{CYC}	320	2000	ns
CLK Low Time	t_1	80		ns
CLK High Time	t_2	120		ns
CLK Rise and Fall Time	t_r, t_f		30	ns
X_1 Rising to CLK Rising	t_{XKR}	30	120	ns
X_1 Rising to CLK Falling	t_{XKF}	30	150	ns
$A_{8\sim15}$ Valid to Leading Edge of Control (1)	t_{AC}	270		ns
$A_{0\sim7}$ Valid to Leading Edge of Control	t_{ACL}	240		ns
$A_{0\sim15}$ Valid to Valid Data In	t_{AD}		575	ns
Address Float After Leading Edge of $\overline{RD}$ ($\overline{INTA}$)	t_{AFR}		0	ns
$A_{8\sim15}$ Valid Before Trailing Edge of ALE(1)	t_{AL}	115		ns
$A_{0\sim7}$ Valid Before Trailing Edge of ALE	t_{ALL}	90		ns
READY Valid from Address Valid	t_{ARY}		220	ns
Address (A_8–A_{15}) Valid After Control	t_{CA}	120		ns
Width of Control Low ($\overline{RD}$, $\overline{WR}$, $\overline{INTA}$)	t_{CC}	400		ns
Trailing Edge of Control to Leading Edge of ALE	t_{CL}	50		ns
Data Valid to Trailing Edge of $\overline{WR}$	t_{DW}	420		ns
HLDA to Bus Enable	t_{HABE}		210	ns
Bus Float After HLDA	t_{HABF}		210	ns

A.C. CHARACTERISTICS cont'd

Parameter	Symbol	80C85A		Units
		Min.	Max.	
HLDA Valid to Trailing Edge of CLK	t_{HACK}	110		ns
HOLD Hold Time	t_{HDH}	0		ns
HOLD Setup Time to Trailing Edge of CLK	t_{HDS}	170		ns
INTR Hold Time	t_{INH}	0		ns
INTR, RST, and TRAP Setup Time to Falling Edge of CLK	t_{INS}	160		ns
Address Hold Time After ALE	t_{LA}	100		ns
Trailing Edge of ALE to Leading Edge of Control	t_{LC}	130		ns
ALE Low During CLK High	t_{LCK}	100		ns
ALE to Valid Data During Read	t_{LDR}		460	ns
ALE to Valid Data During Write	t_{LDW}		200	ns
ALE Width	t_{LL}	140		ns
ALE to READY Stable	t_{LRY}		110	ns
Trailing Edge of $\overline{RD}$ to Re-Enabling of Address	t_{RAE}	150		ns
$\overline{RD}$ (or $\overline{INTA}$) to Valid Data	t_{RD}		300	ns
Control Trailing Edge to Leading Edge of Next Control	t_{RV}	400		ns
Data Hold Time After $\overline{RD}$ $\overline{INTA}$ (7)	t_{RDH}	0		ns
READY Hold Time	t_{RYH}	0		ns
READY Setup Time to Leading Edge of CLK	t_{RYS}	110		ns
Data Valid After Trailing Edge of $\overline{WR}$	t_{WD}	100		ns
LEADING Edge of $\overline{WR}$ to Data Valid	t_{WDL}		40	ns

Notes: (1) A_8–A_{15} address Specs apply to IO/$\overline{M}$, S_0, and S_1 except A_8–A_{15} are undefined during T_4–T_6 of OF cycle whereas IO/$\overline{M}$, S_0, and S_1 are stable.

(2) Test conditions: t_{CYC} = 320ns C_L = 150pF

(3) For all output timing where C_L = 150pF use the following correction factors:
25pF ≤ C_L < 150pF: –0.10ns/pF
150pF < C_L ≤ 300pF: +0.30ns/pF

(4) Output timings are measured with purely capacitive load.

(5) All timings are measured at output voltage V_L = 0.8V, V_H = 2.2V, and 1.5V with 10ns rise and fall time on inputs.

(6) To calculate timing specifications at other values of t_{CYC} use Table 7.

(7) Data hold time is guaranteed under all loading conditions.

Input Waveform for A.C. Tests:

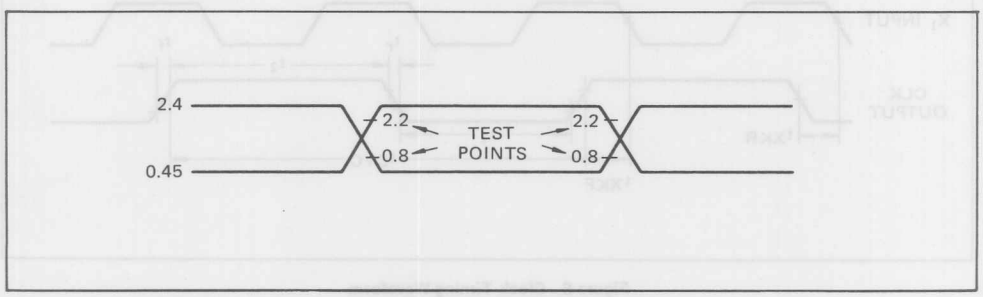


Table 7 Bus Timing Specification as a T_{CYC} Dependent

MSM80C85A			
t_{AL}	–	(1/2)T – 45	MIN
t_{LA}	–	(1/2)T – 60	MIN
t_{LL}	–	(1/2)T – 20	MIN
t_{LCK}	–	(1/2)T – 60	MIN
t_{LC}	–	(1/2)T – 30	MIN
t_{AD}	–	(5/2 + N)T – 225	MAX
t_{RD}	–	(3/2 + N)T – 180	MAX
t_{RAE}	–	(1/2)T – 10	MIN
t_{CA}	–	(1/2)T – 40	MIN
t_{DW}	–	(3/2 + N)T – 60	MIN
t_{WD}	–	(1/2)T – 60	MIN
t_{CC}	–	(3/2 + N)T – 80	MIN
t_{CL}	–	(1/2)T – 110	MIN
t_{ARY}	–	(3/2)T – 260	MAX
t_{HACK}	–	(1/2)T – 50	MIN
t_{HABF}	–	(1/2)T + 50	MAX
t_{HABE}	–	(1/2)T + 50	MAX
t_{AC}	–	(2/2)T – 50	MIN
t_1	–	(1/2)T – 80	MIN
t_2	–	(1/2)T – 40	MIN
t_{RV}	–	(3/2)T – 80	MIN
t_{LDR}	–	(4/2)T – 180	MAX

Note: N is equal to the total WAIT states.
$T = t_{CYC}$

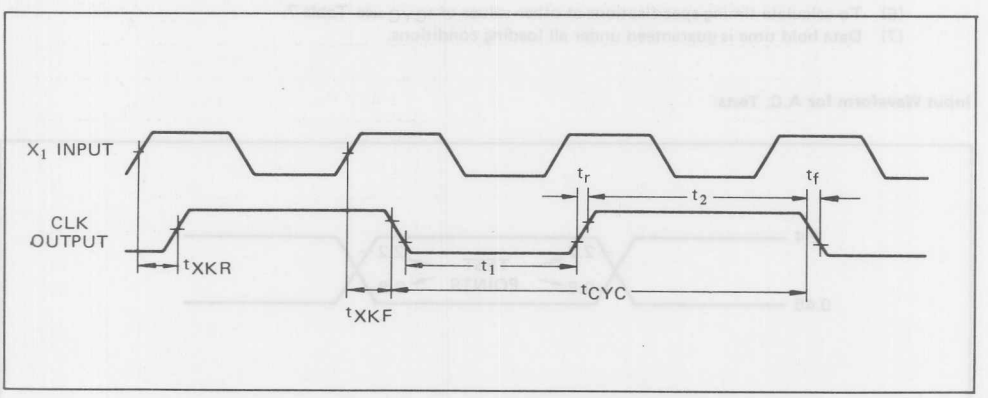
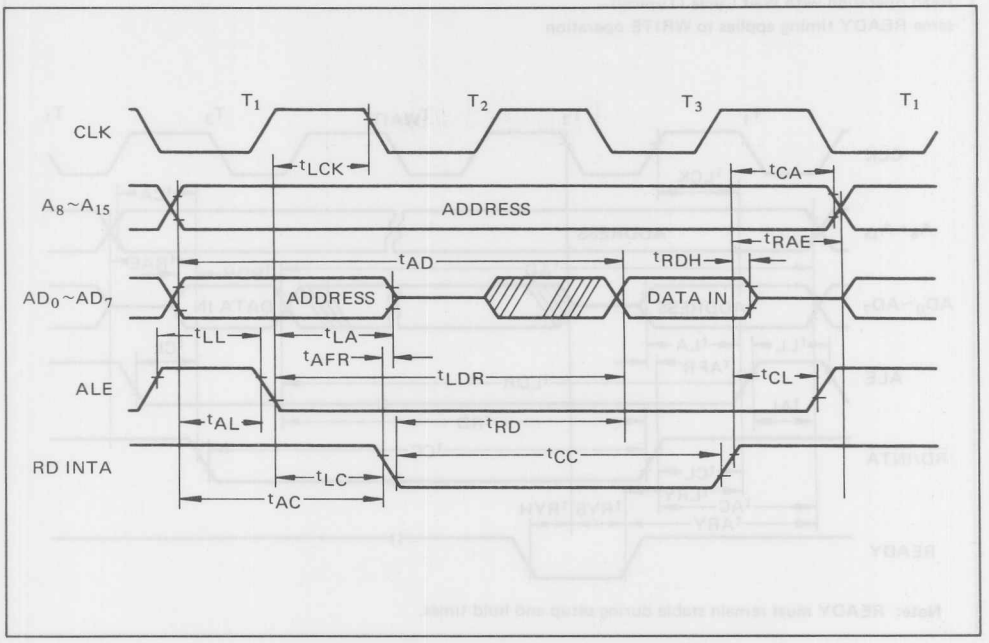
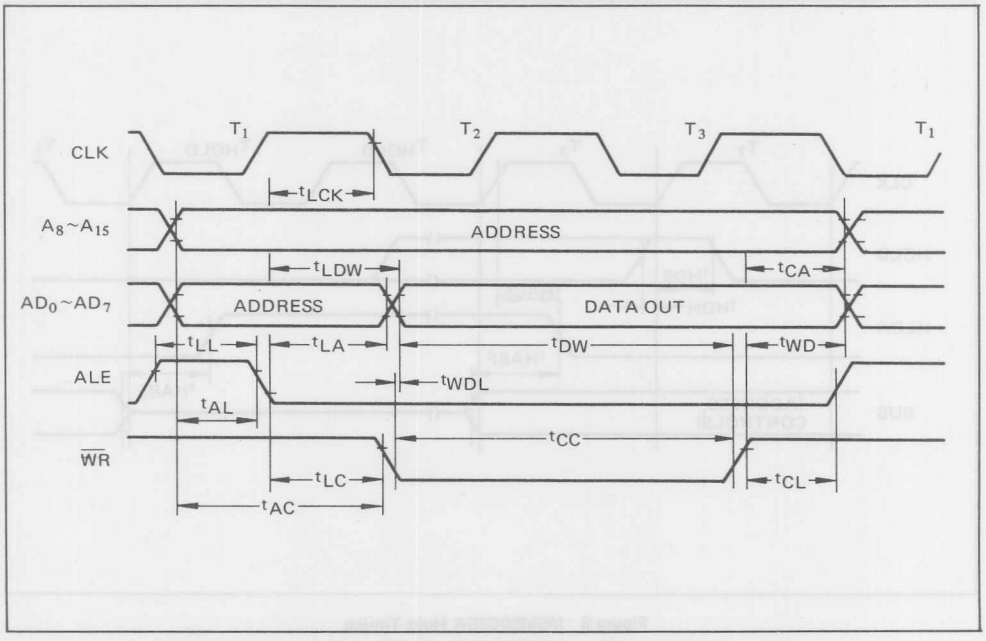


Figure 6 Clock Timing Waveform

READ OPERATION



WRITE OPERATION



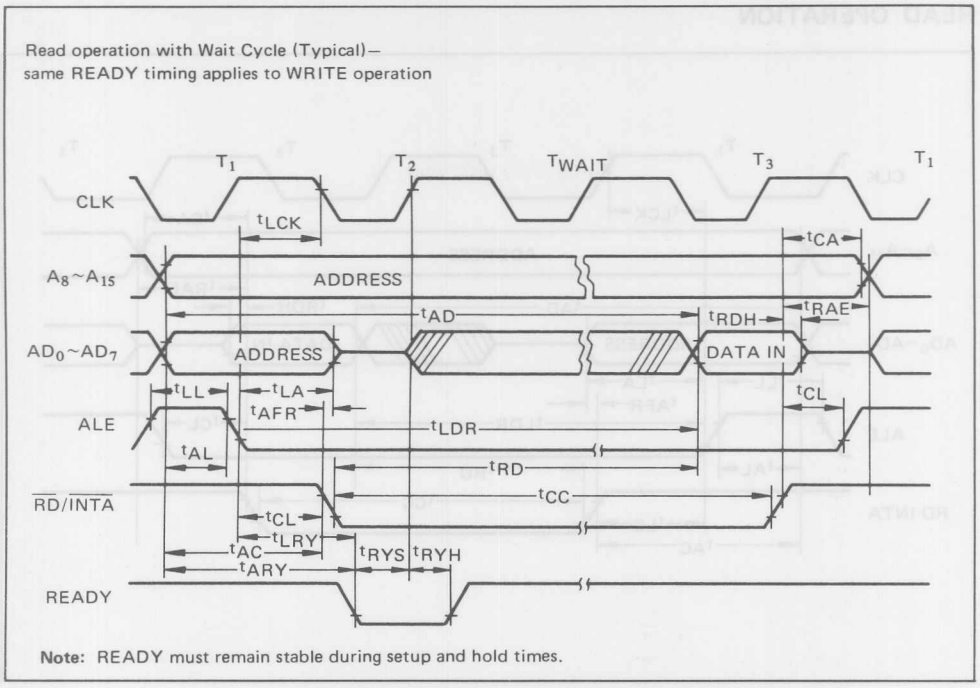


Figure 7 MSM80C85A Bus Timing, With and Without Wait

HOLD OPERATION

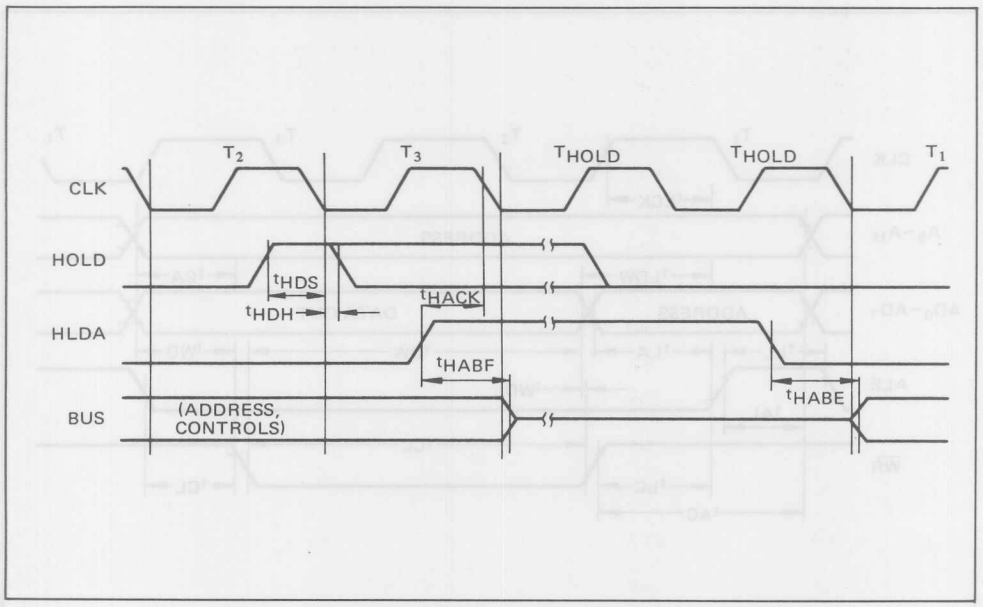
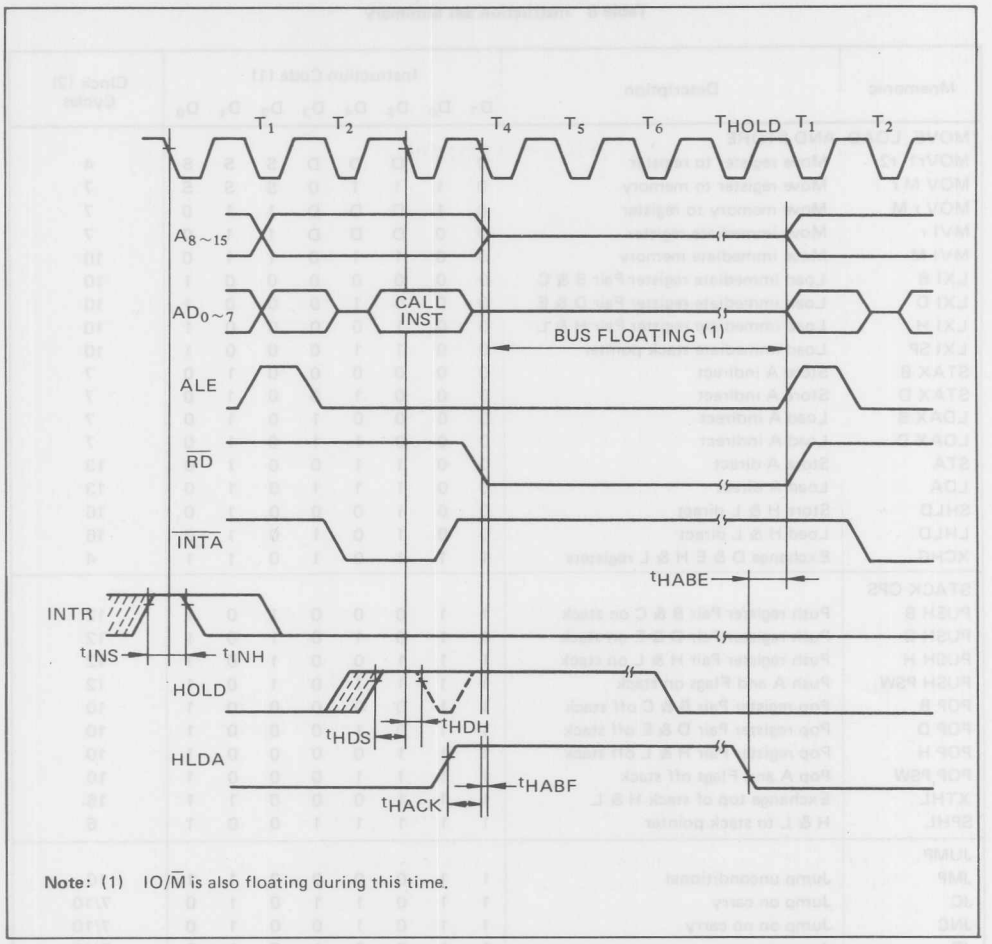


Figure 8 MSM80C85A Hold Timing



Note: (1) IO/$\overline{M}$ is also floating during this time.

Figure 9 MSM80C85A Interrupt and Hold Timing

Mnemonic	Description	Instruction Code (1)								Clock (2) Cycles
		D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	
MOVE, LOAD, AND STORE										
MOVr1 r2	Move register to register	0	1	D	D	D	S	S	S	4
MOV M r	Move register to memory	0	1	1	1	0	S	S	S	7
MOV r M	Move memory to register	0	1	D	D	D	1	1	0	7
MVI r	Move immediate register	0	0	D	D	D	1	1	0	7
MVI M	Move immediate memory	0	0	1	1	0	1	1	0	10
LXI B	Load immediate register Pair B & C	0	0	0	0	0	0	0	1	10
LXI D	Load immediate register Pair D & E	0	0	0	1	0	0	0	1	10
LXI H	Load immediate register Pair H & L	0	0	1	0	0	0	0	1	10
LXI SP	Load immediate stack pointer	0	0	1	1	0	0	0	1	10
STAX B	Store A indirect	0	0	0	0	0	0	1	0	7
STAX D	Store A indirect	0	0	0	1	0	0	1	0	7
LDAX B	Load A indirect	0	0	0	0	1	0	1	0	7
LDAX D	Load A indirect	0	0	0	1	1	0	1	0	7
STA	Store A direct	0	0	1	1	0	0	1	0	13
LDA	Load A direct	0	0	1	1	1	0	1	0	13
SHLD	Store H & L direct	0	0	1	0	0	0	1	0	16
LHLD	Load H & L direct	0	0	1	0	1	0	1	0	16
XCHG	Exchange D & E H & L registers	1	1	1	0	1	0	1	1	4
STACK OPS										
PUSH B	Push register Pair B & C on stack	1	1	0	0	0	1	0	1	12
PUSH D	Push register Pair D & E on stack	1	1	0	1	0	1	0	1	12
PUSH H	Push register Pair H & L on stack	1	1	1	0	0	1	0	1	12
PUSH PSW	Push A and Flags on stack	1	1	1	1	0	1	0	1	12
POP B	Pop register Pair B & C off stack	1	1	0	0	0	0	0	1	10
POP D	Pop register Pair D & E off stack	1	1	0	1	0	0	0	1	10
POP H	Pop register Pair H & L off stack	1	1	1	0	0	0	0	1	10
POP PSW	Pop A and Flags off stack	1	1	1	1	0	0	0	1	10
XTHL	Exchange top of stack H & L	1	1	1	0	0	0	1	1	16
SPHL	H & L to stack pointer	1	1	1	1	1	0	0	1	6
JUMP										
JMP	Jump unconditional	1	1	0	0	0	0	1	1	10
JC	Jump on carry	1	1	0	1	1	0	1	0	7/10
JNC	Jump on no carry	1	1	0	1	0	0	1	0	7/10
JZ	Jump on zero	1	1	0	0	1	0	1	0	7/10
JNZ	Jump on no zero	1	1	0	0	0	0	1	0	7/10
JP	Jump on positive	1	1	1	1	0	0	1	0	7/10
JM	Jump on minus	1	1	1	1	1	0	1	0	7/10
JPE	Jump on parity even	1	1	1	0	1	0	1	0	7/10
JPO	Jump on parity odd	1	1	1	0	0	0	1	0	7/10
PCHL	H & L to program counter	1	1	1	0	1	0	0	1	6
CALL										
CALL	Call unconditional	1	1	0	0	1	1	0	1	18
CC	Call on carry	1	1	0	1	1	1	0	0	9/18
CNC	Call on no carry	1	1	0	1	0	1	0	0	9/18
CZ	Call on zero	1	1	0	0	1	1	0	0	9/18
CNZ	Call on no zero	1	1	0	0	0	1	0	0	9/18
CP	Call on positive	1	1	1	1	0	1	0	0	9/18
CM	Call on minus	1	1	1	1	1	1	0	0	9/18
CPE	Call on parity even	1	1	1	0	1	1	0	0	9/18
CPO	Call on parity odd	1	1	1	0	0	1	0	0	9/18

Table 8 Instruction Set Summary cont'd

Mnemonic	Description	Instruction Code(1) D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Clock(2) Cycles
RETURN										
RET	Return	1	1	0	0	1	0	0	1	10
RC	Return on carry	1	1	0	1	1	0	0	0	6/12
RNC	Return on no carry	1	1	0	1	0	0	0	0	6/12
RZ	Return on zero	1	1	0	0	1	0	0	0	6/12
RNZ	Return on no zero	1	1	0	0	0	0	0	0	6/12
RP	Return on positive	1	1	1	1	0	0	0	0	6/12
RM	Return on minus	1	1	1	1	1	0	0	0	6/12
RPE	Return on parity even	1	1	1	0	1	0	0	0	6/12
RPO	Return on parity odd	1	1	1	0	0	0	0	0	6/12
RESTART										
RST	Restart	1	1	A	A	A	1	1	1	12
INPUT/OUTPUT										
IN	Input	1	1	0	1	1	0	1	1	10
OUT	Output	1	1	0	1	0	0	1	1	10
INCREMENT AND DECREMENT										
INR r	Increment register	0	0	D	D	D	1	0	0	4
DCR r	Decrement register	0	0	D	D	D	1	0	1	4
INR M	Increment memory	0	0	1	1	0	1	0	0	10
DCR M	Decrement memory	0	0	1	1	0	1	0	1	10
INX B	Increment B & C registers	0	0	0	0	0	0	1	1	6
INX D	Increment D & E registers	0	0	0	1	0	0	1	1	6
INX H	Increment H & L registers	0	0	1	0	0	0	1	1	6
INX SP	Increment stack pointer	0	0	1	1	0	0	1	1	6
DCX B	Decrement B & C	0	0	0	0	1	0	1	1	6
DCX D	Decrement D & E	0	0	0	1	1	0	1	1	6
DCX H	Decrement H & L	0	0	1	0	1	0	1	1	6
DCX SP	Decrement stack pointer	0	0	1	1	1	0	1	1	6
ADD										
ADD r	Add register to A	1	0	0	0	0	S	S	S	4
ADC r	Add register to A with carry	1	0	0	0	1	S	S	S	4
ADD M	Add memory to A	1	0	0	0	0	1	1	0	7
ADC M	Add memory to A with carry	1	0	0	0	1	1	1	0	7
ADI	Add immediate to A	1	1	0	0	0	1	1	0	7
ACI	Add immediate to A with carry	1	1	0	0	1	1	1	0	7
DAD B	Add B & C to H & L	0	0	0	0	1	0	0	1	10
DAD D	Add D & E to H & L	0	0	0	1	1	0	0	1	10
DAD H	Add H & L to H & L	0	0	1	0	1	0	0	1	10
DAD SP	Add stack pointer to H & L	0	0	1	1	1	0	0	1	10
SUBTRACT										
SUB r	Subtract register from A	1	0	0	1	0	S	S	S	4
SBB r	Subtract register from A with borrow	1	0	0	1	1	S	S	S	4
SUB M	Subtract memory from A	1	0	0	1	0	1	1	0	7
SBB M	Subtract memory from A with borrow	1	0	0	1	1	1	1	0	7
SUI	Subtract immediate from A	1	1	0	1	0	1	1	0	7
SBI	Subtract immediate from A with borrow	1	1	0	1	1	1	1	0	7

Table 8 Instruction Set Summary cont'd

Mnemonic	Description	Instruction Code (1) D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Clock(2) Cycles
LOGICAL										
ANA r	And register with A	1	0	1	0	0	S	S	S	4
XRA r	Exclusive Or register with A	1	0	1	0	1	S	S	S	4
ORA r	Or register with A	1	0	1	1	0	S	S	S	4
CMP r	Compare register with A	1	0	1	1	1	S	S	S	4
ANA M	And memory with A	1	0	1	0	0	1	1	0	7
XRA M	Exclusive Or memory with A	1	0	1	0	1	1	1	0	7
ORA M	Or memory with A	1	0	1	1	0	1	1	0	7
CMP M	Compare memory with A	1	0	1	1	1	1	1	0	7
ANI	And immediate with A	1	1	1	0	0	1	1	0	7
XRI	Exclusive Or immediate with A	1	1	1	0	1	1	1	0	7
ORI	Or immediate with A	1	1	1	1	0	1	1	0	7
CPI	Compare immediate with A	1	1	1	1	1	1	1	0	7
ROTATE										
RLC	Rotate A left	0	0	0	0	0	1	1	1	4
RRC	Rotate A right	0	0	0	0	1	1	1	1	4
RAL	Rotate A left through carry	0	0	0	1	0	1	1	1	4
RAR	Rotate A right through carry	0	0	0	1	1	1	1	1	4
SPECIALS										
CMA	Complement A	0	0	1	0	1	1	1	1	4
STC	Set carry	0	0	1	1	0	1	1	1	4
CMC	Complement carry	0	0	1	1	1	1	1	1	4
DAA	Decimal adjust A	0	0	1	0	0	1	1	1	4
CONTROL										
EI	Enable Interrupts	1	1	1	1	1	0	1	1	4
DI	Disable Interrupt	1	1	1	1	0	0	1	1	4
NOP	No-operation	0	0	0	0	0	0	0	0	4
HLT	Halt	0	1	1	1	0	1	1	0	5
RIM	Read Interrupt Mask	0	0	1	0	0	0	0	0	4
SIM	Set Interrupt Mask	0	0	1	1	0	0	0	0	4

Notes: (1) DDD or SSS. B 000. C 001. D 010. E 011. H 100. L 101. Memory 110. A 111.
(2) Two possible cycle times, (6/12) indicate instruction cycles dependent on condition flags.

OKI semiconductor

MSM81C55RS/GS

2048 BIT CMOS STATIC RAM WITH I/O PORTS AND TIMER

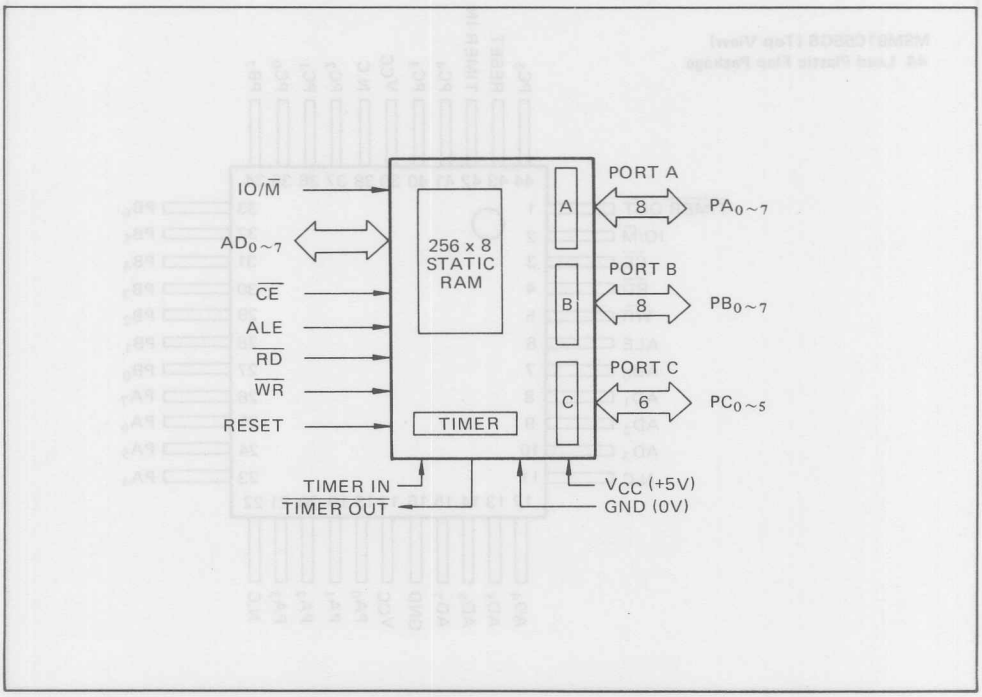
GENERAL DESCRIPTION

The MSM81C55RS/GS is a 2k bit static RAM (256 byte) with parallel I/O ports. It uses silicon gate CMOS technology and consumes a standby current of 100 micro ampere maximum while the chip is not selected. Featuring a maximum access time of 400 ns, the MSM81C55RS/GS can be used in an 80C85A system without using wait states. The parallel I/O consists of two 8-bit ports and one 6-bit port (both general purpose). The MSM81C55RS/GS also contains a 14-bit programmable counter/timer which may be used for sequence-wave generation or terminal count-pulsing.

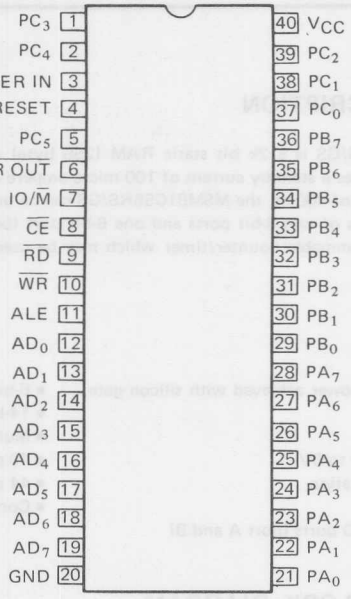
FEATURES

- High speed and low power achieved with silicon gate CMOS technology.
- 256 words x 8 bits
- Single power supply, 3 to 6V
- Completely static operation
- On-chip address latch
- 8-bit programmable I/O ports (port A and B)
- 6-bit programmable I/O port (port C)
- 14-bit programmable binary counter/timer
- Multiplexed address/data bus
- 40 pin DIP package (MSM81C55RS)
- 44 pin flat package (MSM81C55GS)
- Compatible with 8155

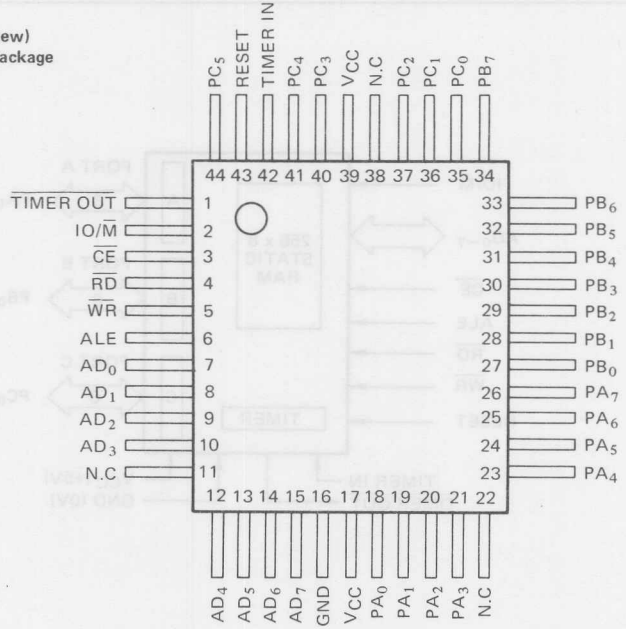
FUNCTIONAL BLOCK DIAGRAM



MSM81C55RS (Top View)
40 Lead Plastic DIP



MSM81C55GS (Top View)
44 Lead Plastic Flap Package



ABSOLUTE MAXIMUM RATINGS

Parameter	Symbol	Conditions	Limits		Unit
			MSM81C55RS	MSM81C55GS	
Supply Voltage	V_{CC}	Referenced to GND	−0.5 to +7		V
Input Voltage	V_{IN}		−0.5 to V_{CC} + 0.5		V
Output Voltage	V_{OUT}		−0.5 to V_{CC} + 0.5		V
Storage Temperature	T_{stg}		−55 to + 150		°C
Allowable Loss	P_D	Ta = 25°C	1.0	0.7	W

OPERATING CONDITION

Parameter	Symbol	Limits	Unit
Supply Voltage	V_{CC}	3 to 6	V
Operating Temperature	T_{OP}	−40 to +85	°C

RECOMMENDED OPERATING CONDITIONS

Parameter	Symbol	Min.	Typ.	Max.	Unit
Supply Voltage	V_{CC}	4.5	5	5.5	V
Operating Temperature	T_{OP}	−40	+25	+85	°C
"L" Level Input	V_{IL}	−0.3		+0.8	V
"H" Level Input	V_{IH}	2.2		V_{CC} + 0.3	V

CHARACTERISTICS

Parameter	Symbol	Conditions		Min.	Typ.	Max.	Unit
"L" Level Output Voltage	V_{OL}	I_{OL} = 2mA	V_{CC} = 4.5V to 5.5V Ta = −40°C to 85°C			0.45	V
"H" Level Output Voltage	V_{OH}	I_{OH} = −400μA		2.4			V
		I_{OH} = −40μA		4.2			V
Input Leak Current	I_{LI}	$0 \leq V_{IN} \leq V_{CC}$		−10		10	μA
Output Leak Current	I_{LO}	$0 \leq V_{OUT} \leq V_{CC}$		−10		10	μA
Standby Current	I_{CCS}	$\overline{CE} \geq V_{CC}-0.2V$ $V_{IH} \geqq V_{CC}-0.2V$ $V_{IL} \leqq V_{CC}-0.2V$			0.1	100	μA
Mean Operating Current	I_{CC}	Memory cycle time: 1μs				5	mA

AC CHARACTERISTICS

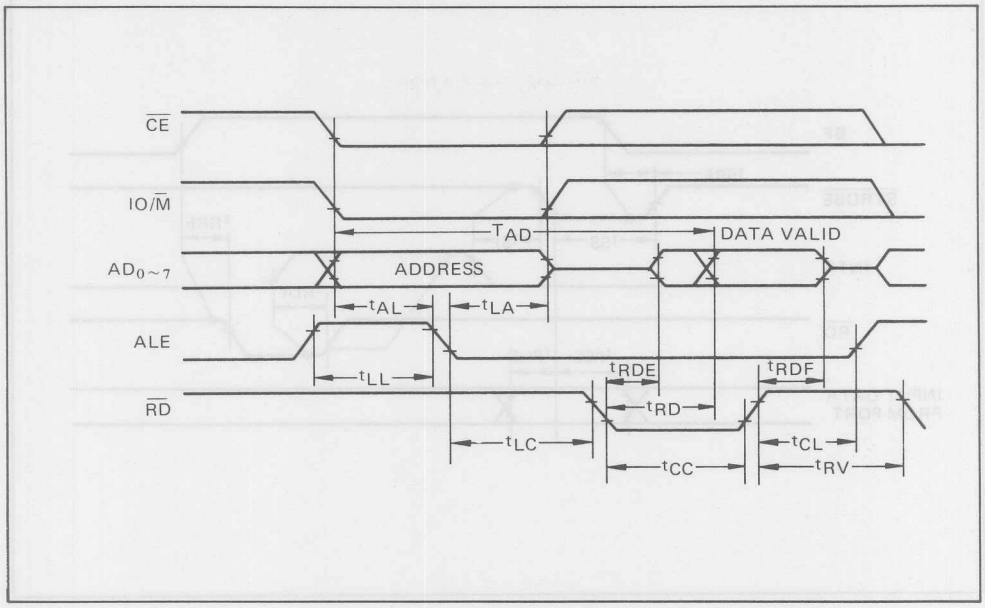
(V_{CC} = 4.5 to 5.5V, Ta = –40 to +85°C)

Parameter	Symbol	Min.	Max.	Unit	Remarks
Address/latch Set-up Time	t_{AL}	50		ns	
Latch/address Hold Time	t_{LA}	30		ns	
Latch/read (write) Delay Time	t_{LC}	100		ns	
Read/output Delay Time	t_{RD}		170	ns	
Address/output Delay Time	t_{AD}		400	ns	
Latch Width	t_{LL}	100		ns	
Read/data Bus Floating Time	t_{RDF}	0	100	ns	
Read (write)/latch Delay Time	t_{CL}	20		ns	
Read (write) Width	t_{CC}	250		ns	
Data In/write Set-up Time	t_{DW}	150		ns	
Write/data-in Hold Time	t_{WD}	0		ns	
Recovery Time	t_{RV}	300		ns	
Write/port Output Delaty Time	t_{WP}		400	ns	
Port Input/read Set-up Time	t_{PR}	70		ns	Load capacitance: 150pF
Read/port Input Hold Time	t_{RP}	50		ns	
Strobe/buffer Full Delay Time	t_{SBF}		400	ns	
Strobe Width	t_{SS}	200		ns	
Strobe/buffer Empty Delay Time	t_{RBE}		400	ns	
Strobe/interrupt-on delay time	t_{SI}		400	ns	
Read/interrupt-off Delay Time	t_{RDI}		400	ns	
Port Input/strobe Set-up Time	t_{PSS}	50		ns	
Strobe/port-input Hold Time	t_{PHS}	120		ns	
Strobe/buffer-empty Delay Time	t_{SBE}		400	ns	
Write/buffer-full Delay Time	t_{WBF}		400	ns	
Write/interrupt-off Delay Time	t_{WI}		400	ns	
Timer Output Delay Time	t_{TL}		400	ns	
Timer Output Delay Time	t_{TH}		400	ns	
Read/data Bus Enable Delay Time	t_{RDE}	10		ns	
Timer Cycle Time	t_{CYC}	320		ns	
Timer Input Rise and Fall Times	t_r, t_f		80	ns	
Timer Input Low Level Time	t_1	80		ns	
Timer Input High Level Time	t_2	120		ns	

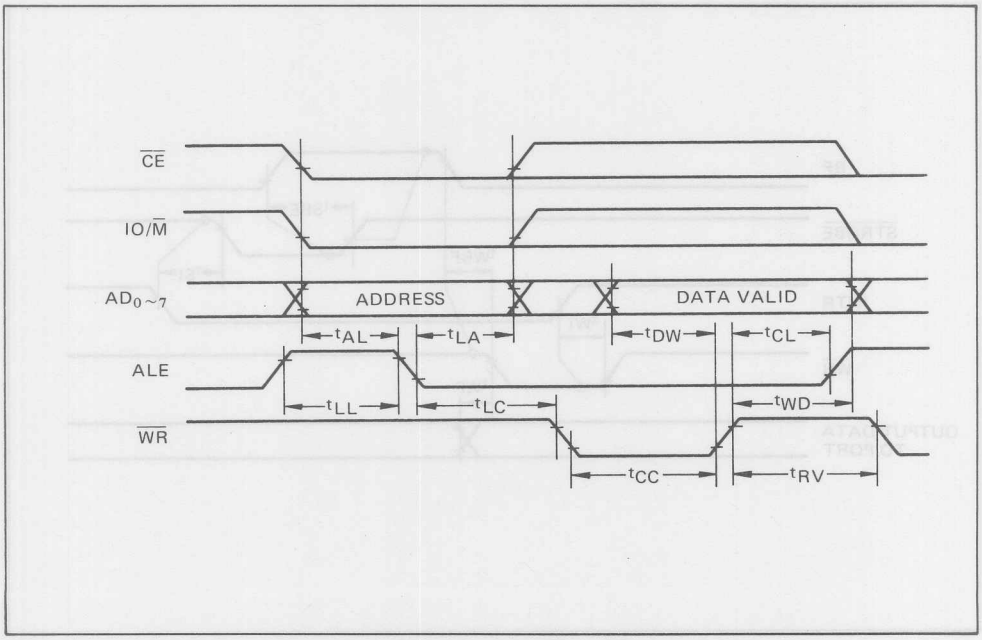
Note: Timing are measured with V_L = 0.8V and V_H = 2.2V for both input and output.

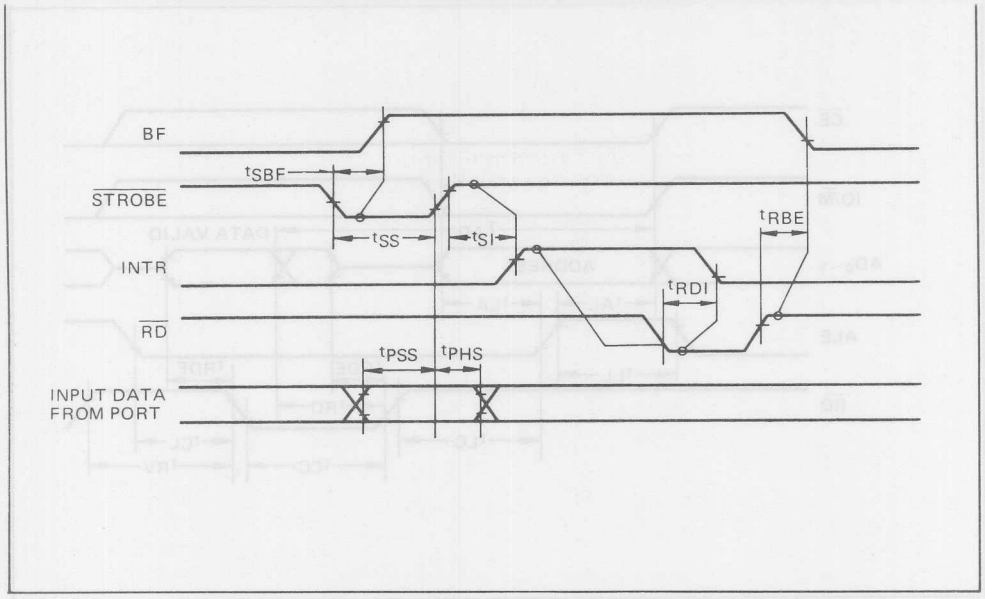
TIMING

Read Cycle

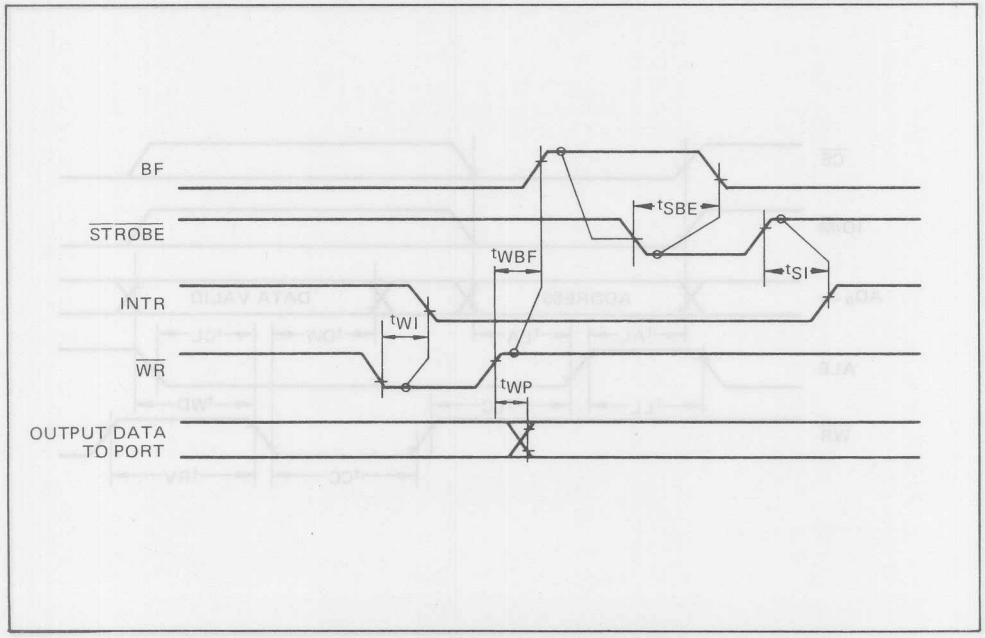


Write Cycle

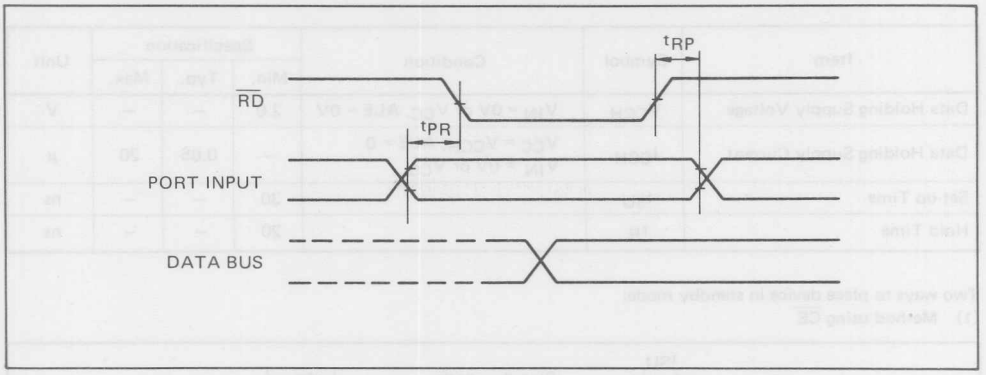




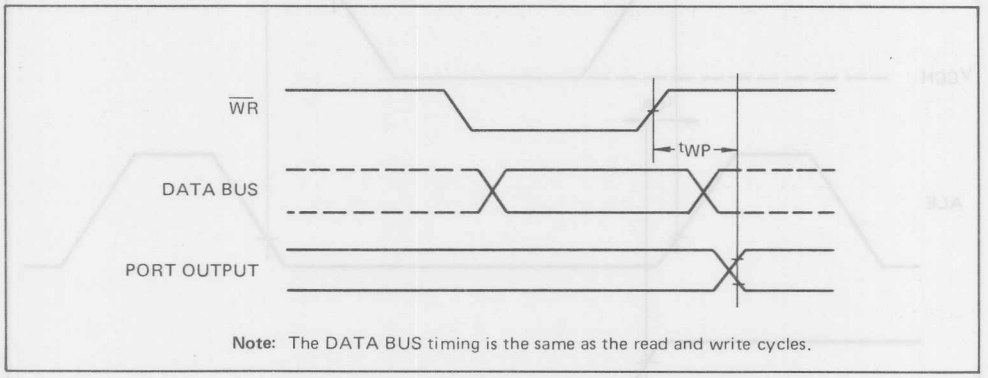
Strobe Output Mode



Basic Input Mode

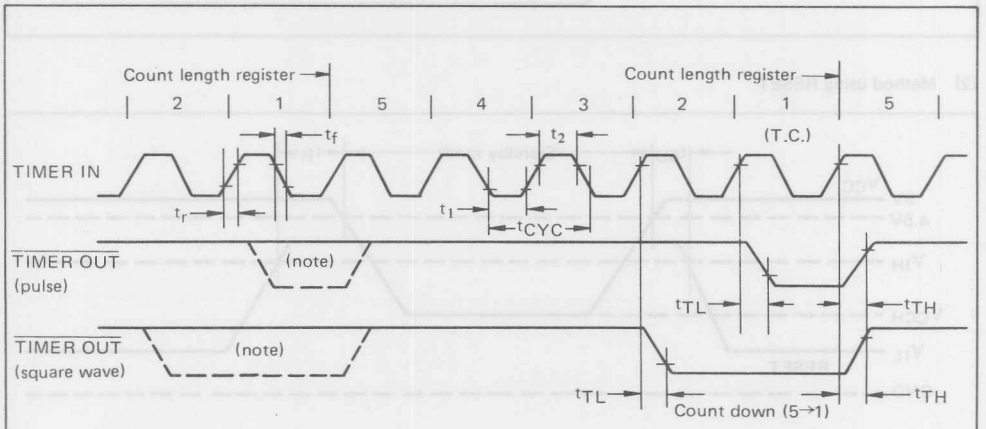


Basic Output Mode



Note: The DATA BUS timing is the same as the read and write cycles.

Timer Waveforms

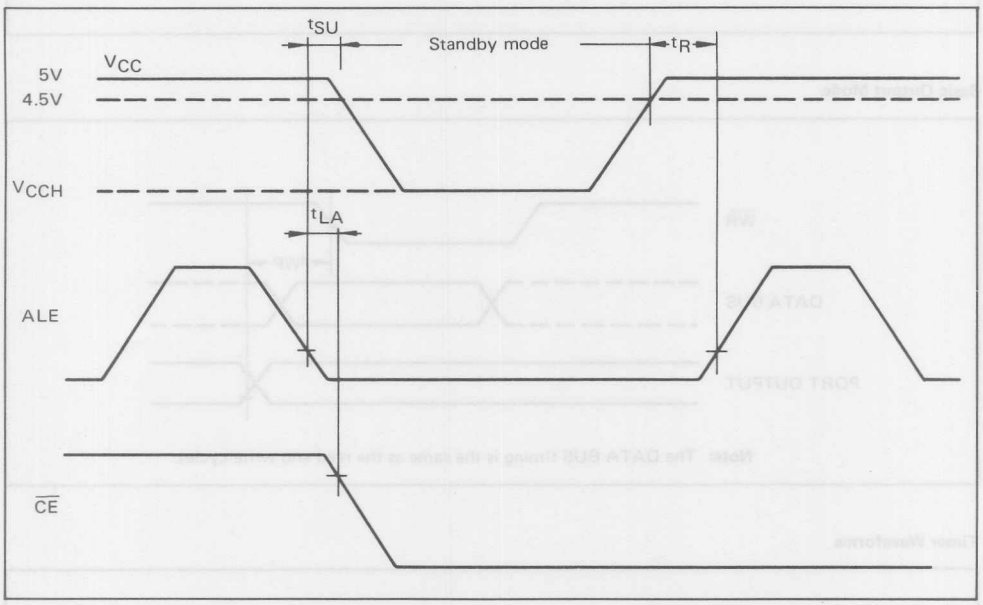


Note: Periodically output according to the output mode (m1=1) programming contents.

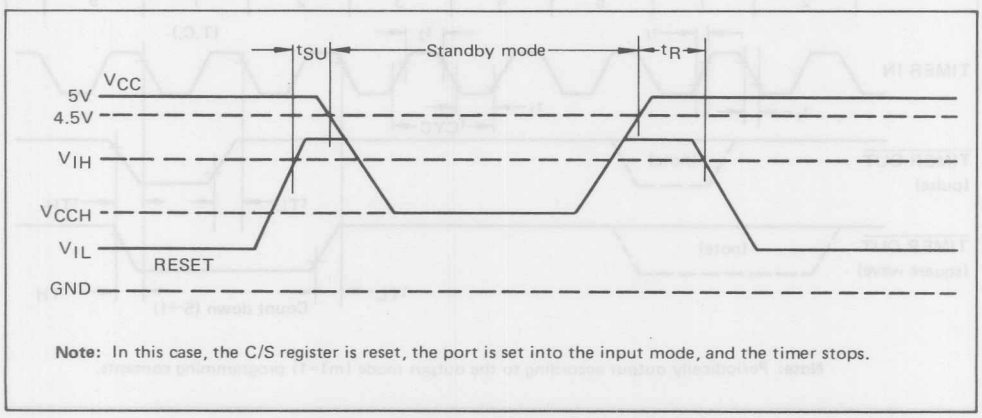
RAM DATA HOLD CHARACTERISTICS AT LOW SUPPLY VOLTAGE

Item	Symbol	Condition	Specification			Unit
			Min.	Typ.	Max.	
Data Holding Supply Voltage	V_{CCH}	V_{IN} = 0V or V_{CC}, ALE = 0V	2.0	–	–	V
Data Holding Supply Current	I_{CCH}	$V_{CC} = V_{CCH}$, ALE = 0 V_{IN} = 0V or V_{CC}	–	0.05	20	μ
Set-up Time	t_{SU}		30	–	–	ns
Hold Time	t_R		20	–	–	ns

Two ways to place device in standby mode:
(1) Method using $\overline{CE}$



(2) Method using RESET



Note: In this case, the C/S register is reset, the port is set into the input mode, and the timer stops.

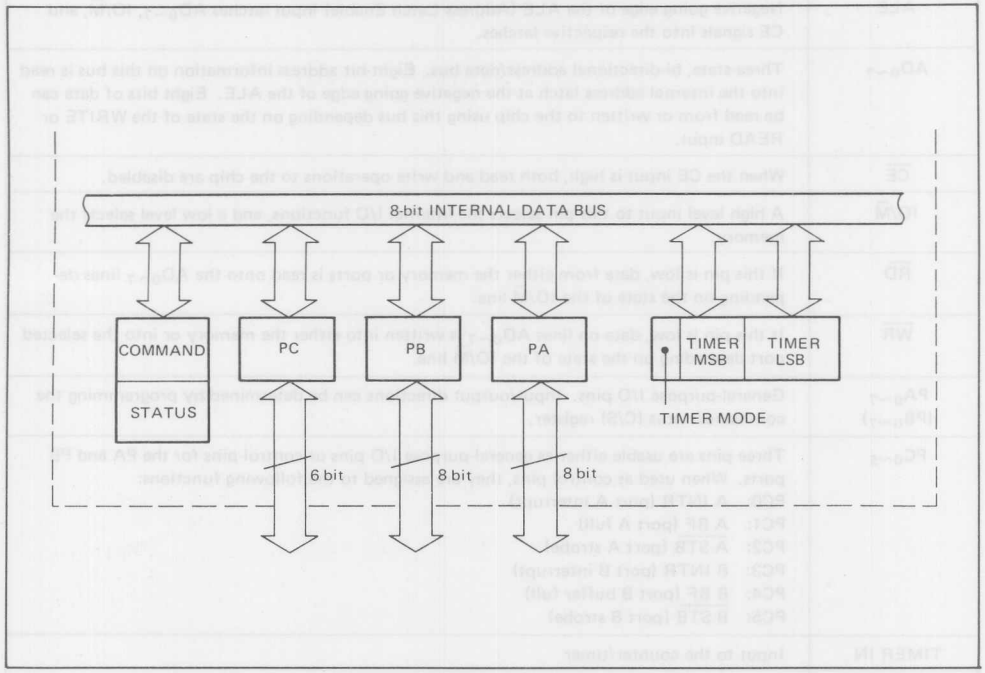
PIN FUNCTIONS

Symbol	Function
RESET	A high level input to this pin resets the chip, placing all three I/O ports in the input mode, and stops timer.
ALE	Negative going edge of the ALE (Address Latch Enable) input latches $AD_{0\sim7}$, IO/$\overline{M}$, and CE signals into the respective latches.
$AD_{0\sim7}$	Three-state, bi-directional address/data bus. Eight-bit address information on this bus is read into the internal address latch at the negative going edge of the ALE. Eight bits of data can be read from or written to the chip using this bus depending on the state of the WRITE or READ input.
$\overline{CE}$	When the CE input is high, both read and write operations to the chip are disabled.
IO/$\overline{M}$	A high level input to this pin selects the internal I/O functions, and a low level selects the memory.
$\overline{RD}$	If this pin is low, data from either the memory or ports is read onto the $AD_{0\sim7}$ lines depending on the state of the IO/$\overline{M}$ line.
$\overline{WR}$	It this pin is low, data on lines $AD_{0\sim7}$ is written into either the memory or into the selected port depending on the state of the IO/$\overline{M}$ line.
$PA_{0\sim7}$ ($PB_{0\sim7}$)	General-purpose I/O pins. Input/output directions can be determined by programming the command/status (C/S) register.
$PC_{0\sim5}$	Three pins are usable either as general-purpose I/O pins or control pins for the PA and PB ports. When used as control pins, they are assigned to the following functions: PC0: A INTR (port A interrupt) PC1: A BF (port A full) PC2: $\overline{A\ STB}$ (port A strobe) PC3: B INTR (port B interrupt) PC4: B BF (port B buffer full) PC5: $\overline{B\ STB}$ (port B strobe)
TIMER IN	Input to the counter/timer
$\overline{TIMER\ OUT}$	Timer output. When the present count is reached during timer operation, this pin provides a square-wave or pulse output depending on the programmed control status.
V_{CC}	3—6V power supply
GND	GND

below.

- 2K bit static RAM (256 words x 8 bits)

and the I/O addresses are described in the table below.

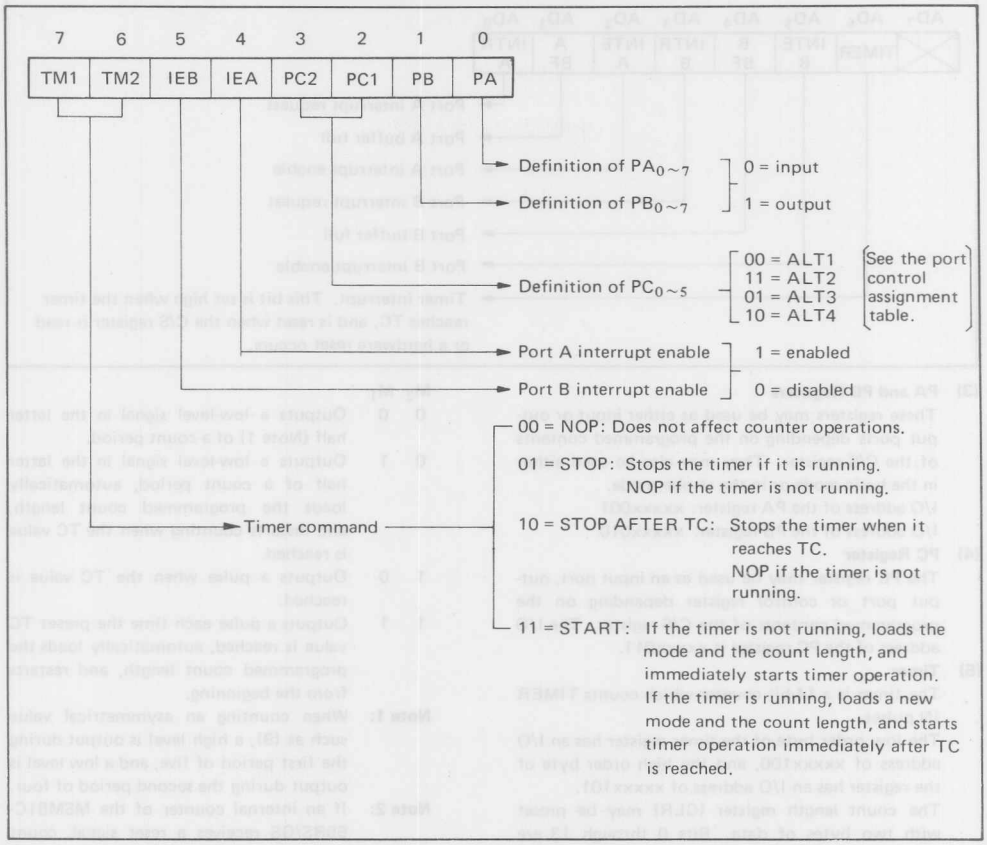


I/O Address								Selecting Register
A7	A6	A5	A4	A3	A2	A1	A0	
x	x	x	x	x	0	0	0	Internal command/status register
x	x	x	x	x	0	0	1	Universal I/O port A (PA)
x	x	x	x	x	0	1	0	Universal I/O port B (PB)
x	x	x	x	x	0	1	1	I/O port C (PC)
x	x	x	x	x	1	0	0	Timer count lower position 8 bits (LSB)
x	x	x	x	x	1	0	1	Timer count upper position 6 bits and timer mode 2 bits (MSB)

x: Don't care.

(1) Programming the Command/Status (C/S) Register

The contetns of the command register can be written during an I/O cycle by addressing it with an I/O address of xxxxx000. Bit assignments for the register are shown below:

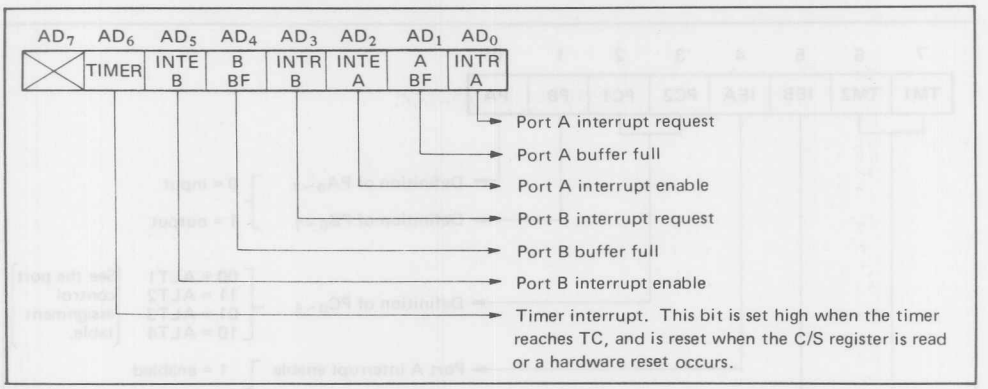


Port Control Assignment Table

Pin	ALT1	ALT2	ALT3	ALT4
PC_0	Input port	Output port	A INTR	A INTR
PC_1	Input port	Output port	A BF	A BF
PC_2	Input port	Output port	A STB	A STB
PC_3	Input port	Output port	Output port	B INTR
PC_4	Input port	Output port	Output port	B BF
PC_5	Input port	Output port	Output port	B STB

(2) Reading the C/S Register

The I/O and timer status can be accessed by reading the contents of the Status register located at I/O address xxxxx000. The status word format is shown below:



(3) PA and PB Registers

These registers may be used as either input or output ports depending on the programmed contents of the C/S register. They may also be used either in the basic mode or in the strobe mode.

I/O address of the PA register: xxxxx001

I/O address of the PB register: xxxxx010

(4) PC Register

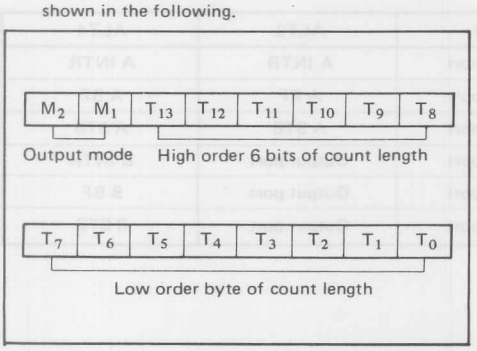
The PC register may be used as an input port, output port or control register depending on the programmed contents of the C/S register. The I/O address of the PC register is xxxxx011.

(5) Timer

The timer is a 14-bit counter which counts TIMER IN pulses.

The low order byte of the timer register has an I/O address of xxxxx100, and the high order byte of the register has an I/O address of xxxxx101.

The count length register (CLR) may be preset with two bytes of data. Bits 0 through 13 are assigned to the count length and bits 14 and 15 specify the timer output mode. A read operation of the CLR reads the contents of the counter and the pertinent output mode. The initial value range which can initially be loaded into the counter is 2 through 3FFF hex. Bit assignments to the timer counter and possible output modes are shown in the following.



M_2	M_1	
0	0	Outputs a low-level signal in the latter half (Note 1) of a count period.
0	1	Outputs a low-level signal in the latter half of a count period, automatically loads the programmed count length, and restarts counting when the TC value is reached.
1	0	Outputs a pulse when the TC value is reached.
1	1	Outputs a pulse each time the preset TC value is reached, automatically loads the programmed count length, and restarts from the beginning.

Note 1: When counting an asymmetrical value such as (9), a high level is output during the first period of five, and a low level is output during the second period of four.

Note 2: If an internal counter of the MSM81C-55RS/GS receives a reset signal, count operation stops but the counter is not set to a specific initial value or output mode. When restarting count operation after reset, the START command must be executed again through the C/S register.

(6) Standby Mode (see page 7)

The MSM81C55RS/GS is placed in standby mode when the high level at $\overline{CE}$ input is latched during the negative going edge of ALE. All input ports and the timer input should be pulled up or down to either V_{CC} or GND potential.

When using battery back-up, all ports should be set low or in input port mode. The timer output should be set low. Otherwise, a buffer should be added to the timer output and the battery should be connected to the power supply pins of the buffer.

By setting the reset input to a high level, the standby mode can be selected. In this case, the command register is reset, so the ports automatically set to the input mode and the timer stops.

OKI semiconductor

MSM82C12RS/GS

8-BIT INPUT/OUTPUT PORT

GENERAL DESCRIPTION

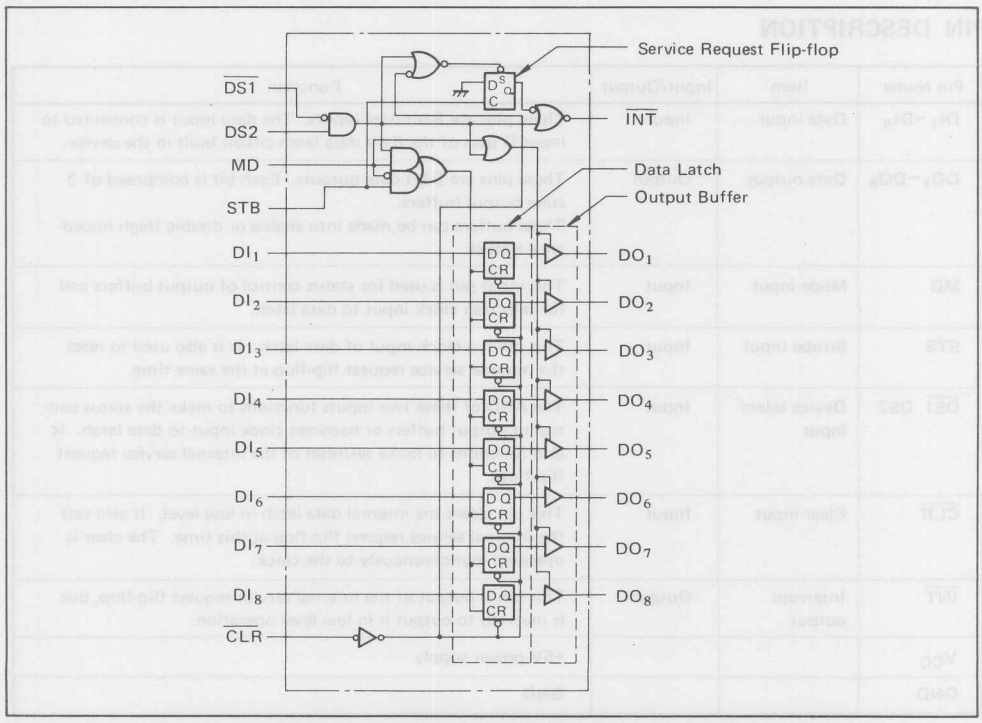
MSM82C12 is an 8-bit input/output port employing 3 μ silicon gate CMOS technology. It operates on a lower power supply. This device incorporates service request flip-flop for generation and control of interrupts for CPU, in addition to a 8-bit latch circuit having a three-state output buffer.

Especially, it is effective when used as an address latch device which separates the time division bus line output from CPU in the system employing MSM80C85A into address bus line and data bus line.

FEATURES

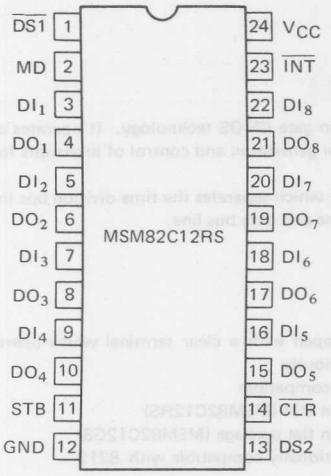
- Operated on a low power supply owing to the adoption of silicon gate CMOS.
- 3 V ~ 6 V single power supply
- Full static operation
- Parallel 8-bit data register and buffer
- Provided with interrupt generating function by the adoption of the service request flip-flop
- Equipped with a clear terminal which operates asynchronously
- TTL compatible
- 24-pin DIP (MSM82C12RS)
- 24-pin flat package (MSM82C12GS)
- Functionally compatible with 8212

CIRCUIT CONFIGURATION

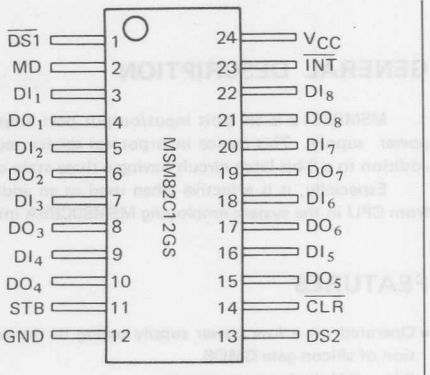


PIN CONFIGURATION

MSM82C12RS (Top View) 24 Lead Plastic DIP



MSM82C12GS (Top View) 24 Lead Plastic Flat Package



A simplified form is used in some cases for model name indication on the device.

PIN DESCRIPTION

Pin Name	Item	Input/Output	Function
DI_1~DI_8	Data input	Input	These pins are 8-bit data inputs. The data input is connected to input D pins of the 8-bit data latch circuit built in the device.
DO_1~DO_8	Data output	Output	These pins are 8-bit data outputs. Each bit is composed of 3-state output buffers. These buffers can be made into enable or disable (high impedance status).
MD	Mode input	Input	This input pin is used for status control of output buffers and for selection clock input to data latch.
STB	Strobe input	Input	This pin is a clock input of data latch. It is also used to reset the internal service request flip-flop at the same time.
$\overline{DS1}$, DS2	Device select input	Input	The AND of these two inputs functions to make the status control of output buffers or becomes clock input to data latch. It also functions to make set/reset of the internal service request flip-flop.
$\overline{CLR}$	Clear input	Input	This pin clears the internal data latch in low level. It also sets the internal service request flip-flop at this time. The clear is operated asynchronously to the clock.
$\overline{INT}$	Interrupt output	Output	This pin is output of the internal service request flip-flop, but is inverted to output it in low level operation.
V_{CC}			+5V power supply
GND			GND

FUNCTIONAL DESCRIPTION

Output Buffer Status Control and Data Latch Clock Input

When the input MD is in high level, the output buffer is enabled and the device select input (DS1.DS2) becomes clock input to data latch. When the input MD is in low level, the status of output buffer is determined by the device select input (DS1.DS2) (the output buffer is enabled when (DS1.DS2) is in high level). At this time, the input STB becomes clock input to data latch.

MD	$(\overline{DS1} \cdot DS2)$	STB	$DO_1 \sim DO_8$
0	0	0	High impedance status
0	0	1	High impedance status
1	0	0	Data latch
1	0	1	Data latch
0	1	0	Data latch
0	1	1	Data in
1	1	0	Data in
1	1	1	Data in

Service Request Flip-flop

The service request flip-flop is used to generate and control interrupt for CPU when the MSM82C12 is used as input/output port in a microcomputer system. The flip-flop is set asynchronously by input CLR. When the flip-flop is set, the system is in non-interrupt status.

$\overline{CLR}$	$(\overline{DS1} \cdot DS2)$	STB	Q	$\overline{INT}$
0	0	0	1	1
0	1	0	1	0
1	1	↓	1	0
1	1	0	1	0
1	0	0	1	1
1	0	↓	0	0

Clear

When the clear input becomes low level, the internal data latch is cleared irrespective of clock and it become low level.

ABSOLUTE MAXIMUM RATINGS

Parameter	Symbol	Conditions	Limits		Unit
			MSM82C12RS	MSM82C12GS	
Supply Voltage	V_{CC}	With respect to GND	−0.5 to +7		V
Input Voltage	V_{IN}		−0.5 to V_{CC}+0.5		V
Output Voltage	V_{OUT}		−0.5 to V_{CC}+0.5		V
Storage Temperature	T_{stg}		−55 to +150		°C
Permissible Loss	P_D	Ta = 25°C	0.9	0.7	W

OPERATING RANGE

Parameter	Symbol	Limits	Unit
Supply Voltage	V_{CC}	3 to 6	V
Operating Temperature	T_{OP}	−40 to +85	°C

RECOMMENDED OPERATING CONDITION

Parameter	Symbol	Min.	Typ.	Max.	Unit
Supply Voltage	V_{CC}	4.5	5	5.5	V
Operating Temperature	T_{OP}	−40	+25	+85	°C
"L" Input Voltage	V_{IL}	−0.3		+0.8	V
"H" Input Voltage	V_{IH}	2.2		V_{CC} +0.3	V

"L" Output Voltage	V_{OL}	I_{OL} = 4mA	V_{CC} = −4.5V to 5.5V Ta = −40°C to +85°C			0.4	V
"H" Output Voltage	V_{OH}	I_{OH} = −4mA		3.7			V
Input Leak Current	I_{LI}	$0 \leqq V_{IN} \leqq V_{CC}$		−10		10	μA
Output Leak Current	I_{LO}	$0 \leqq V_{OUT} \leqq V_{CC}$		−10		10	μA
Supply Current (Standby)	I_{CCS}	$V_{IH} \geqq V_{CC} - 0.2V$ $V_{IL} \leqq 0.2V$			0.1	100	μA
Average Supply Current (active)	I_{CC}	f = 1 MHz				1	mA

AC CHARACTERISTICS

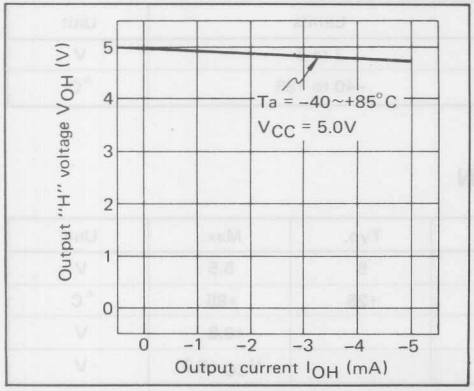
(V_{CC} = 4.5 ~ 5.5V, Ta = −40°C +85°C)

Parameter	Symbol	Min.	Typ.	Max.	Unit	Remarks
Pulse Width	t_{PW}	30			ns	Load 30pF
Data to Output Delay	t_{PD}		20	45	ns	
Write Enable to Output Delay	t_{WE}		31	60	ns	
Data Set Up Time	t_{SET}	15			ns	
Data Hold Time	t_H	30			ns	
Clear to Output Delay	t_C		19	40	ns	
Reset to Output Delay	t_R		21	45	ns	
Set to Output Delay	t_S		25	45	ns	
Output Enable Time	t_E		52	90	ns	Load 20pF + 1 kΩ
Output Disable Time	t_D		30	55	ns	

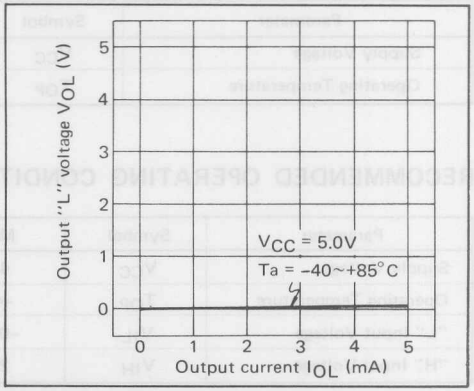
Note: TYP is measured where V_{CC} = 5 V and Ta = 25°C.
Timing is measured where VL = VH = 1.5V in both input and output.
t_E and t_D are measured at V_{OL} + 0.5V or V_{OH} − 0.5V when the two are made into high impedance status.

OUTPUT CHARACTERISTICS (DC Characteristics Reference Value)

(1) Output "H" voltage (V_{OH}) vs. output current (I_{OH})



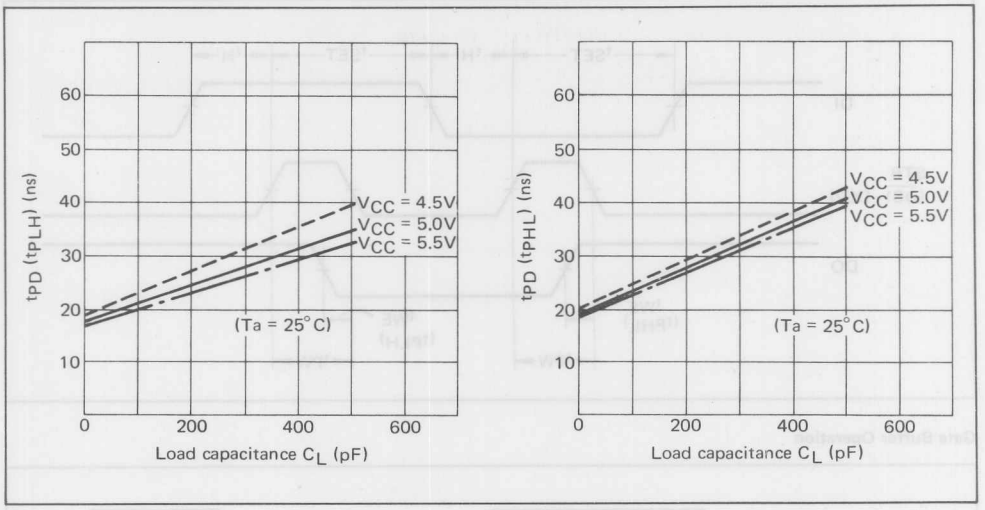
(2) Output "L" voltage (V_{OL}) vs. output current (I_{OL})



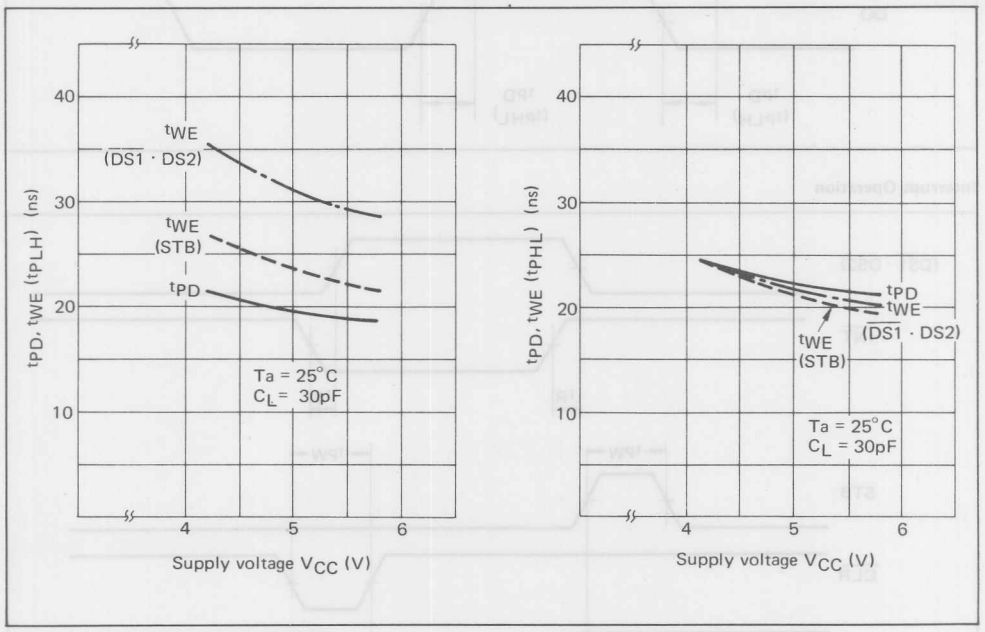
Note: The direction of flowing in is taken as positive for output current.

OUTPUT CHARACTERISTICS (AC Characteristics Reference Value)

(1) t_{PD} vs. load capacitance

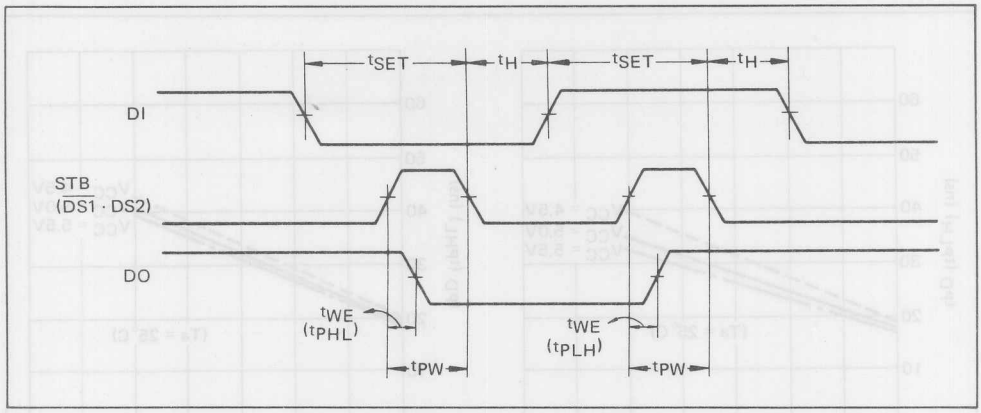


(2) t_{PD} and t_{WE} vs. supply voltage

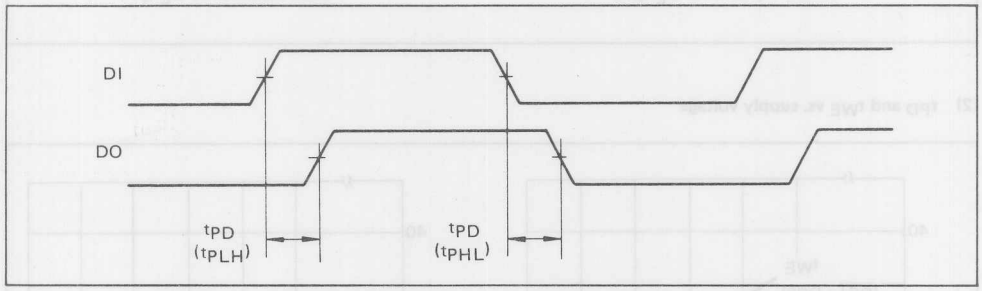


TIMING CHART

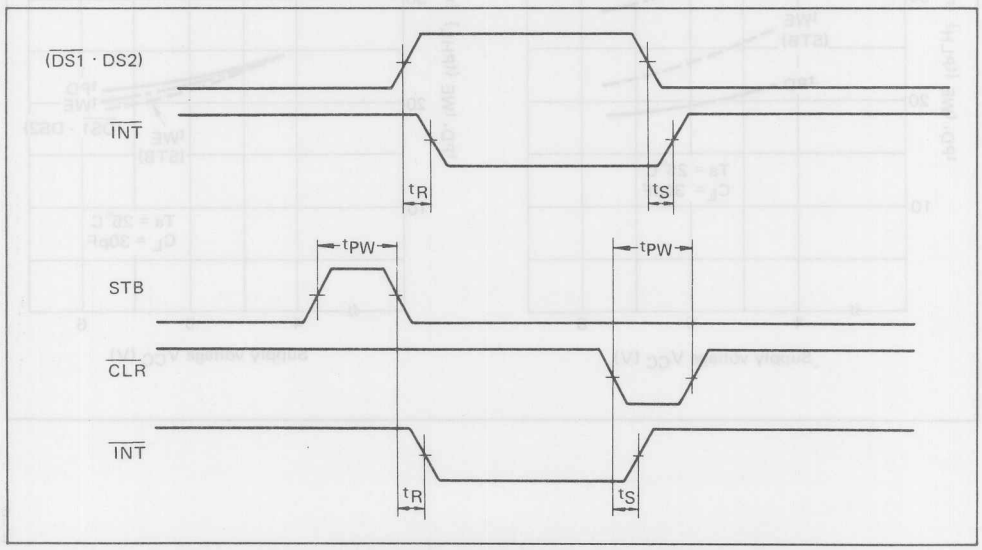
Data Latch Operation



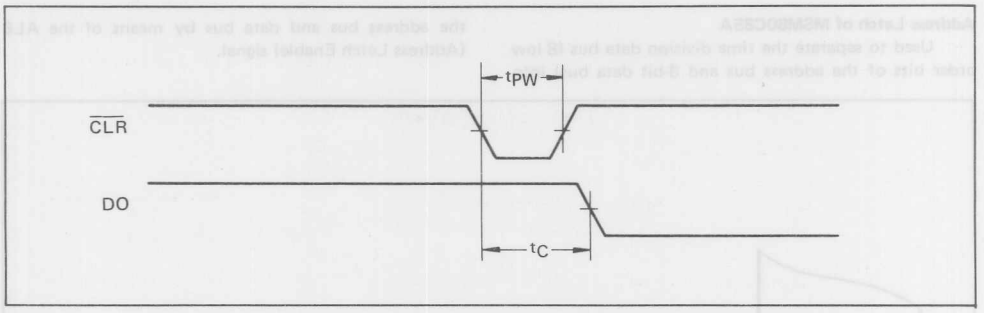
Gate Buffer Operation



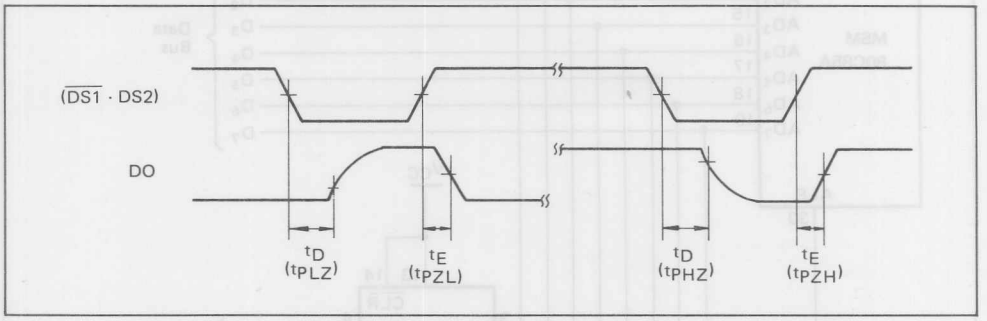
Interrupt Operation



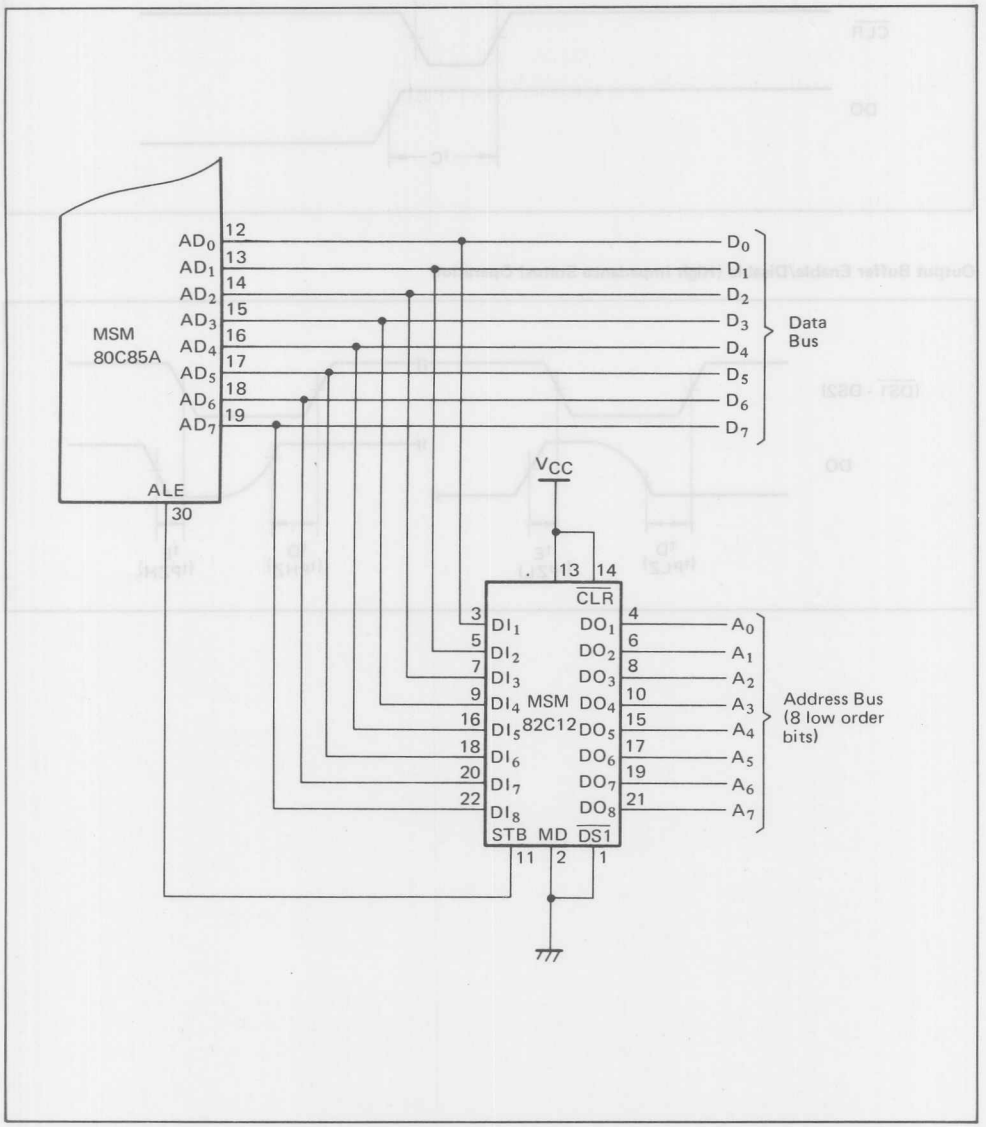
Clear Operation



Output Buffer Enable/Disable (High Impedance Status) Operation



order bits of the address bus and 8-bit data bus) into



Preliminary

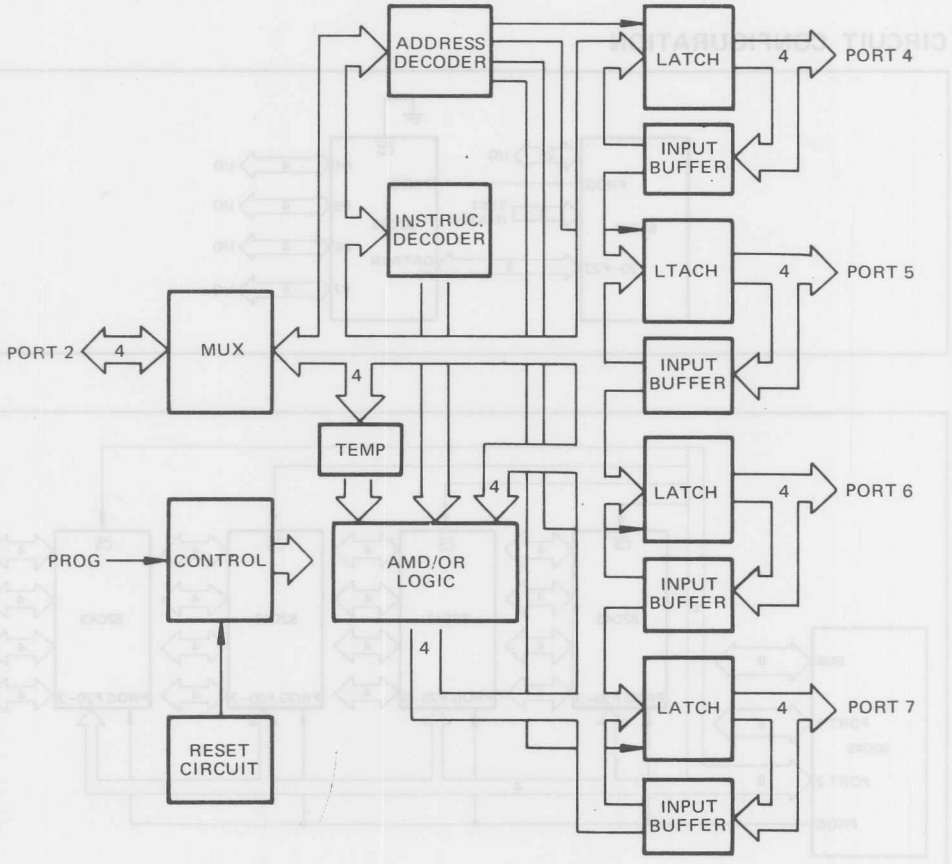
MSM82C43RS

8 BIT I/O PORT

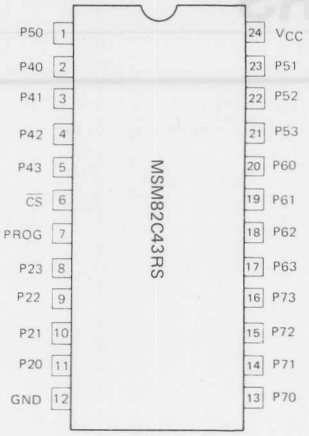
GENERAL DESCRIPTION

FEATURES

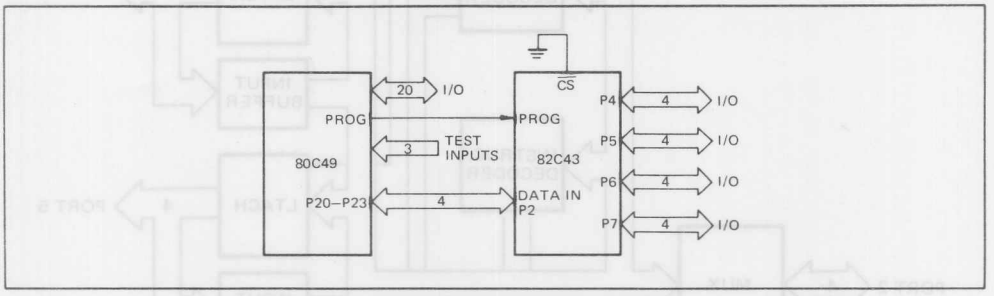
FUNCTIONAL BLOCK DIAGRAM

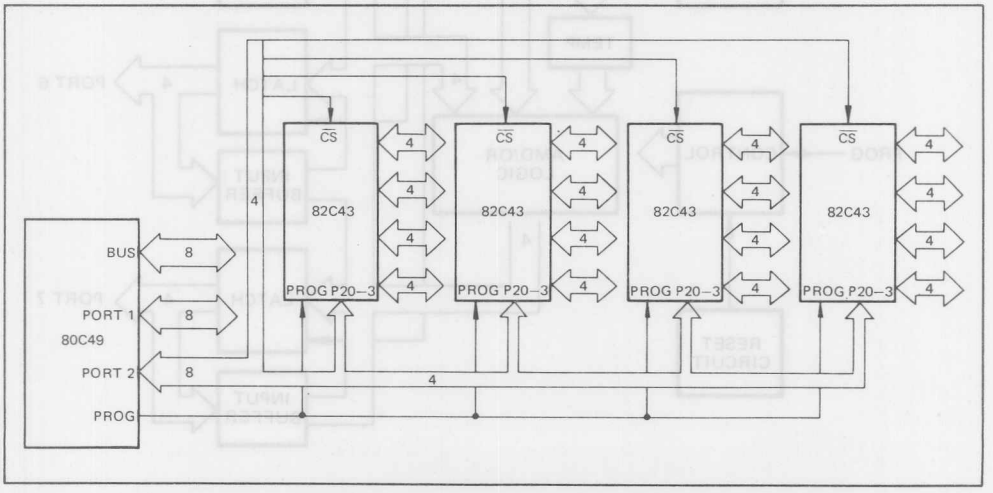


PIN CONFIGURATION



CIRCUIT CONFIGURATION





ABSOLUTE MAXIMUM RATINGS

Parameter	Symbol	Conditions	Limits	Unit
Supply Voltage	V_{CC}	Ta = 25°C	−0.3 to 7	V
Input Voltage	V_I	Ta = 25°C	−0.3 to V_{CC}	V
Storage Temperature	Tstg		−55 to +150	°C

OPERATING CONDITIONS

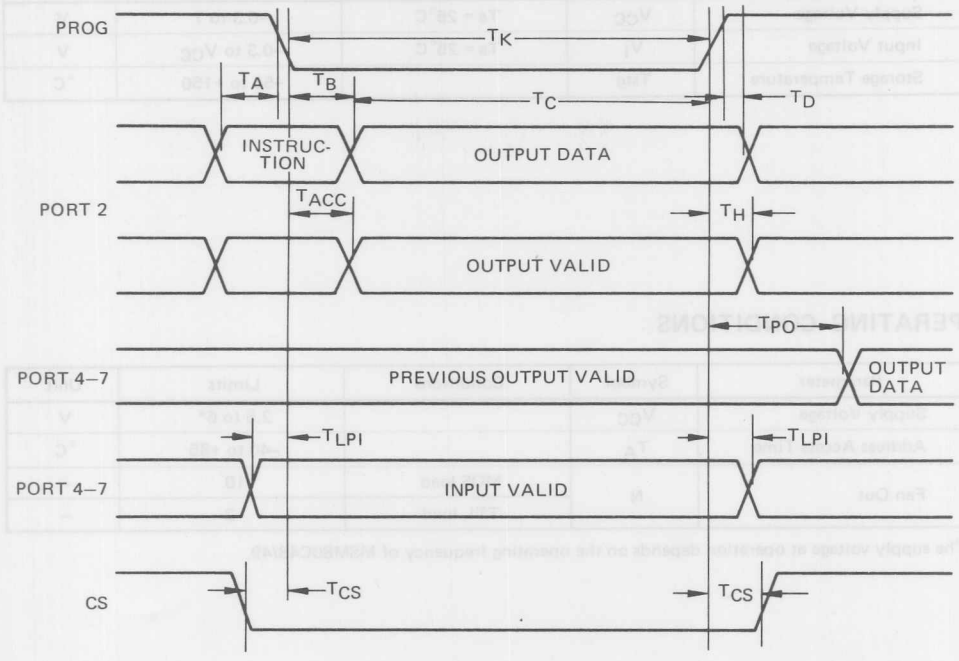
Parameter	Symbol	Conditions	Limits	Unit
Supply Voltage	V_{CC}		2.5 to 6*	V
Address Access Time	T_A		−40 to +85	°C
Fan Out	N	MOS load	10	—
		TTL load	3	—

* The supply voltage at operation depends on the operating frequency of MSM80C48/49.

AC CHARACTERISTICS

(V_{CC} = 5V ±10%, Ta = −40 to +85°C)

Parameter	Symbol	Conditions	Min.	Max.	Unit
Port Control Set Time (to the falling edge of $\overline{PROG}$)	T_A	80PF LOAD	100		ns
Port Control Hold Time (from the rising edge of $\overline{PROG}$)	T_B	20PF LOAD	60		ns
Output Data Set Time	T_C	80PF LOAD	200		ns
Output Data Hold Time	T_D	20PF LOAD	20		ns
Input Data Hold Time	T_H	20PF LOAD	0	150	ns
$\overline{PROG}$ Pulse Width	T_K		700		ns
CS Valid Time (before and after PROG)	T_{CS}		50		ns
Output Data Valid Time (at ports 4 ~ 7)	T_{PO}	100PF LOAD		700	ns
Input Data Hold Time (at ports 4 ~7)	T_{LPI}		100		ns
Input Data Valid Time (from the falling edge of $\overline{PROG}$)	T_{ACC}	80PF LOAD		650	ns



DC CHARACTERISTICS

(V_{CC} = 5V ±10%, Ta = –40 to +85° C)

Parameter		Symbol	Conditions	Min.	Typ.	Max.	Unit
"L" Input Voltage		V_{IL}		–0.5		0.8	V
"H" Input Voltage		V_{IH}		2.0		V_{CC}+0.5	V
"H" Input Voltage		V_{IH}					V
"L" Output Voltage	Ports4–7	V_{OL1}	I_{OL} = 5 mA			0.45	V
"L" Output Voltage	Port7	V_{OL2}	I_{OL} = 20 mA			1	V
"L" Output Voltage	Port2	V_{OL3}	I_{OL} = 0.6 mA			0.45	V
"H" Output Voltage	Ports4–7	V_{OH1}	I_{OH} = 240 μA	2.4			V
"H" Output Voltage	Port2	V_{OH2}	I_{OH} = 100 μA	2.4			V
"H" Output Voltage	Ports4–7	V_{OH1}	I_{OH} = 20 μA	4.2			V
"H" Output Voltage	Port2	V_{OH2}	I_{OH} = 10 μA	4.2			V
Input Leak Current	Ports4–7	I_{L1}	$V_{SS} \leqq V_{IN} \leqq V_{CC}$	–10		20	μ
Input Leak Current	Port2, PROG, CS	I_{L2}	$V_{SS} \leqq V_{IN} \leqq V_{CC}$	–10		10	μ
"L" Output Leak Current	Ports4–7	I_{LO}		2.4			m
Power Supply Current		I_{CC}	Under steady state		5	100	μ

OKI semiconductor

MSM82C51ARS/GS

UNIVERSAL SYNCHRONOUS ASYNCHRONOUS RECEIVER TRANSMITTER

GENERAL DESCRIPTION

MSM82C51A is USART(Universal Synchronous Asynchronous Receiver Transmitter) for serial data communication developed for the microcomputer system.

As a peripheral device of the microcomputer system, MSM82C51A receives parallel data from CPU and transmits serial data after conversion. This device also receives serial data from outside and transmits parallel data to CPU after conversion. Thus the device is used for serial data communication.

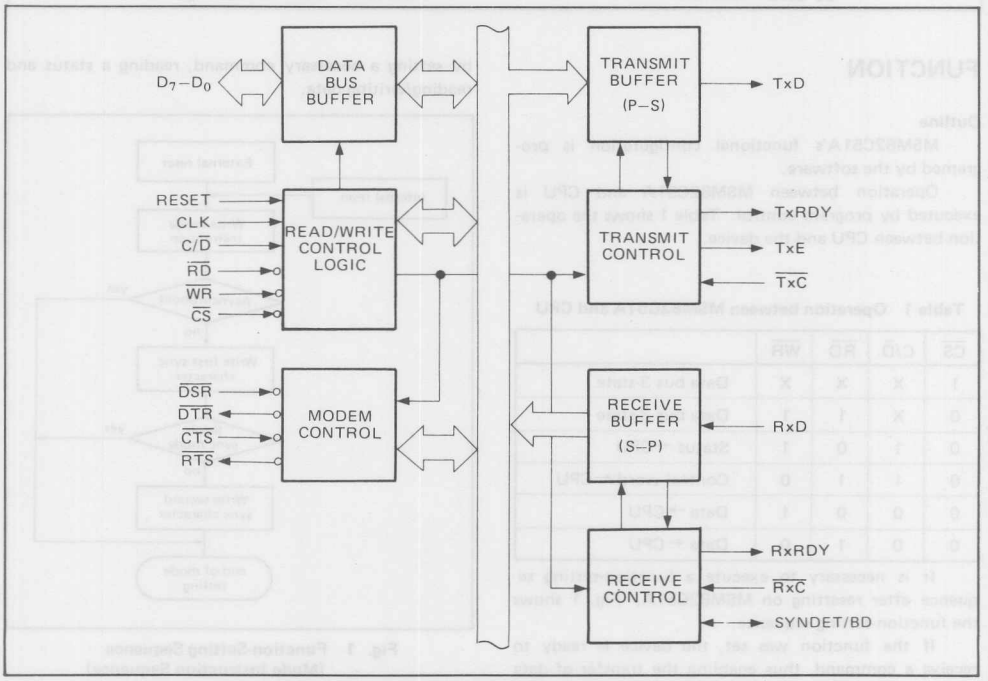
MSM82C51A configures a fully static circuit using silicon gate CMOS technology. Therefore, it operates on an extremely low power supply at 100 μA (max) of standby current by suspending all the operations.

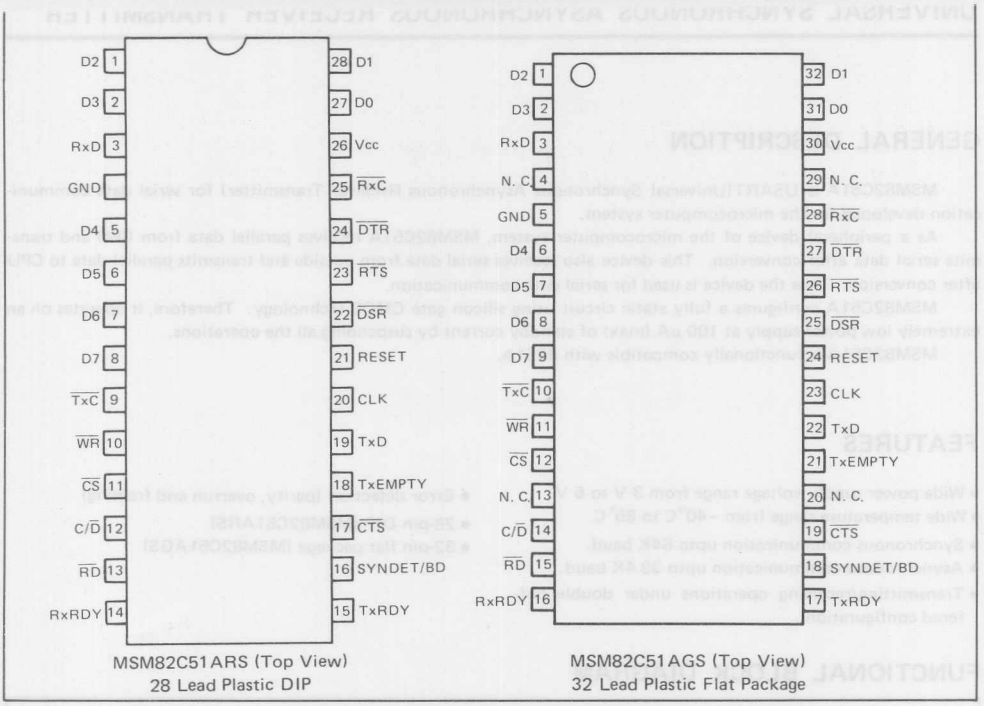
MSM82C51A is functionally compatible with 8251A.

FEATURES

- Wide power supply voltage range from 3 V to 6 V.
- Wide temperature range from −40°C to 85°C.
- Synchronous communication upto 64K baud.
- Asynchronous communication upto 38.4K baud.
- Transmitting/receiving operations under double buffered configuration.
- Error detection (parity, overrun and framing)
- 28-pin DIP (MSM82C51ARS)
- 32-pin flat package (MSM82C51AGS)

FUNCTIONAL BLOCK DIAGRAM





MSM82C51ARS (Top View)
28 Lead Plastic DIP

MSM82C51AGS (Top View)
32 Lead Plastic Flat Package

FUNCTION

Outline

MSM82C51A's functional configuration is programed by the software.

Operation between MSM82C51A and CPU is executed by program control. Table 1 shows the operation between CPU and the device.

Table 1 Operation between MSM82C51A and CPU

$\overline{CS}$	C/$\overline{D}$	$\overline{RD}$	$\overline{WR}$	
1	X	X	X	Data bus 3-state
0	X	1	1	Data bus 3-state
0	1	0	1	Status → CPU
0	1	1	0	Control word ← CPU
0	0	0	1	Data → CPU
0	0	1	0	Data ← CPU

It is necessary to execute a function-setting sequence after resetting on MSM82C51A. Fig. 1 shows the function-setting sequence.

If the function was set, the device is ready to receive a command, thus enabling the transfer of data by setting a necessary command, reading a status and reading/writing data.

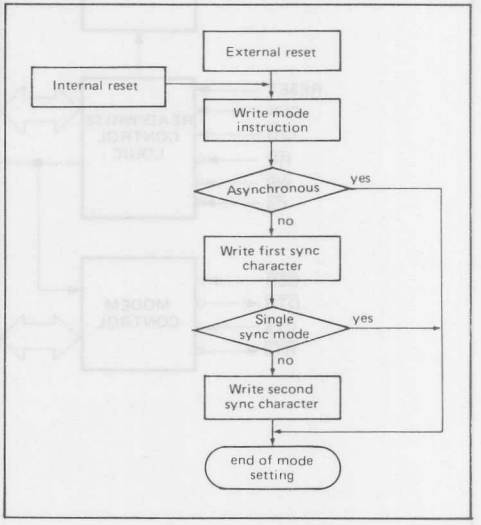


Fig. 1 Function-Setting Sequence (Mode Instruction Sequence)

Control Words

There are two types of control word.

1. Mode instruction (setting of function)
2. Command (setting of operation)

1) Mode Instruction

Mode instruction is used for setting the function of MSM82C51A. Mode instruction will be in "wait for write" at either internal reset or external reset. That is, the writing of control word after resetting will be recognized as "mode instruction."

Items to be set by mode instruction are as follows:

- Synchronous/asynchronous mode
- Stop bit length (asynchronous mode)
- Character length
- Parity bit
- Baud rate factor (asynchronous mode)
- Internal/external synchronization (synchronous mode)
- No. of synchronous characters (synchronous mode)

The bit configuration of mode instruction is shown in Fig.'s 2 and 3. In the case of synchronous mode, it is necessary to write one- or two-tyte sync characters.

If sync characters were written, a function will be set because the writing of sync characters constitutes part of mode instruction.

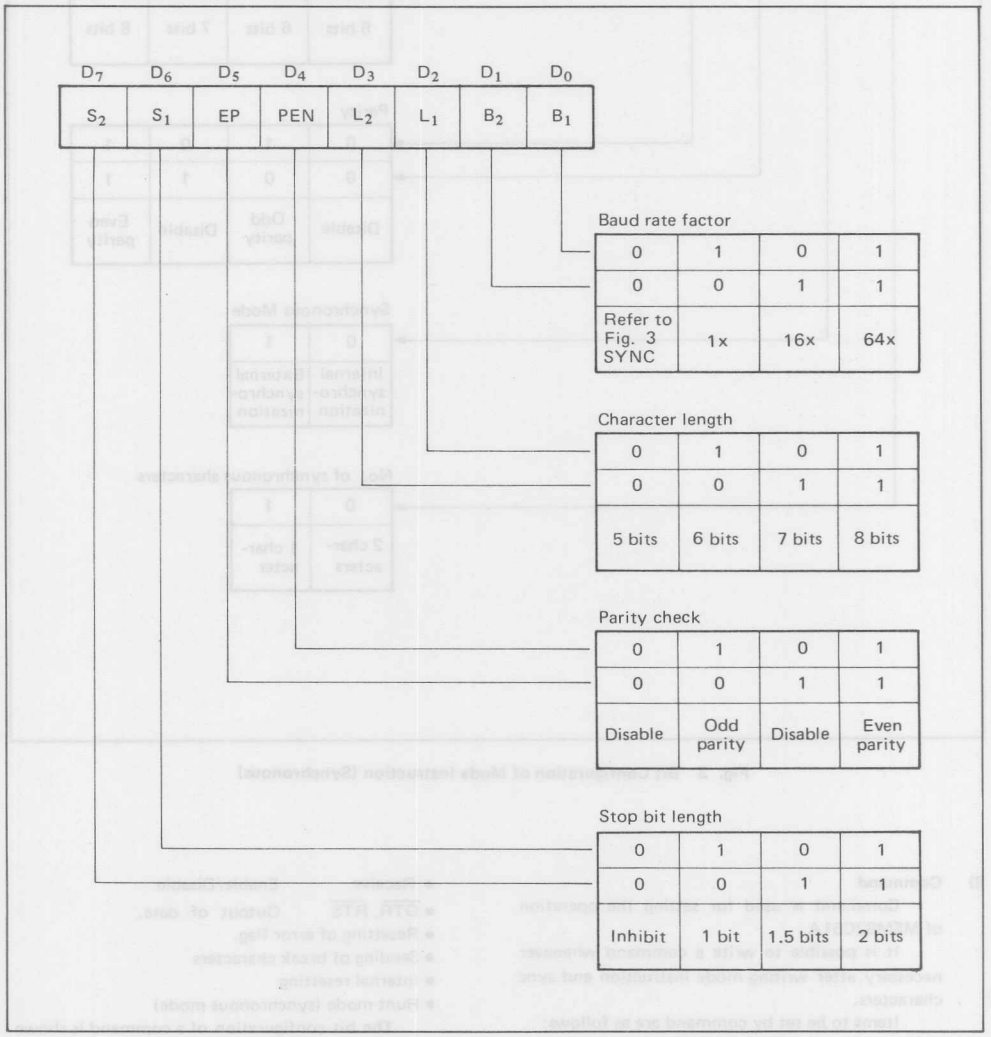


Fig. 2 Bit Configuration of Mode Instruction (Asynchrnous)

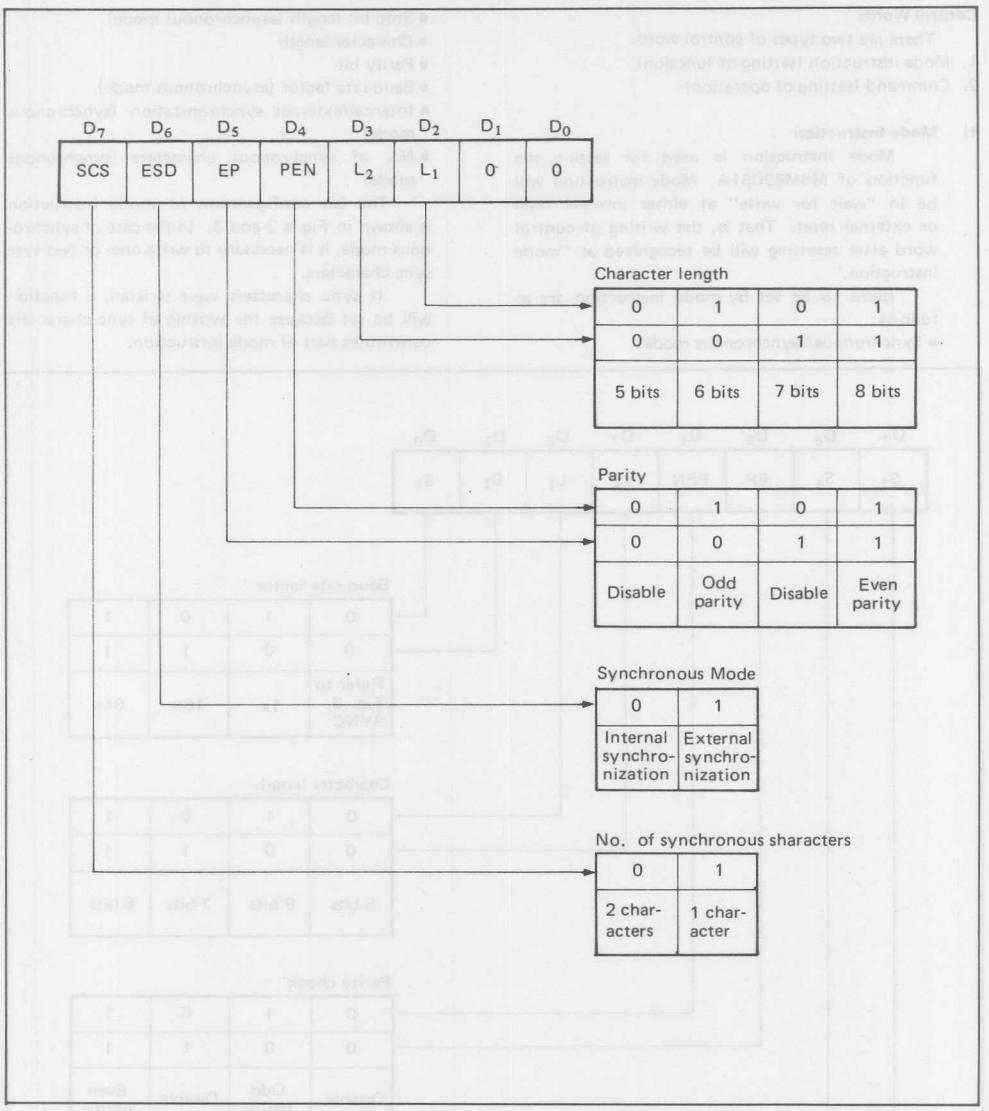


Fig. 3 Bit Configuration of Mode Instruction (Synchronous)

2) Command

Command is used for setting the operation of MSM82C51A.

It is possible to write a command whenever necessary after writing mode instruction and sync characters.

Items to be set by command are as follows:

- Transmit Enable/Disable
- Receive Enable/Disable
- $\overline{DTR}$, $\overline{RTS}$ Output of data.
- Resetting of error flag.
- Sending of break characters
- Internal resetting
- Hunt mode (synchronous mode)

The bit configuration of a command is shown in Fig. 4.

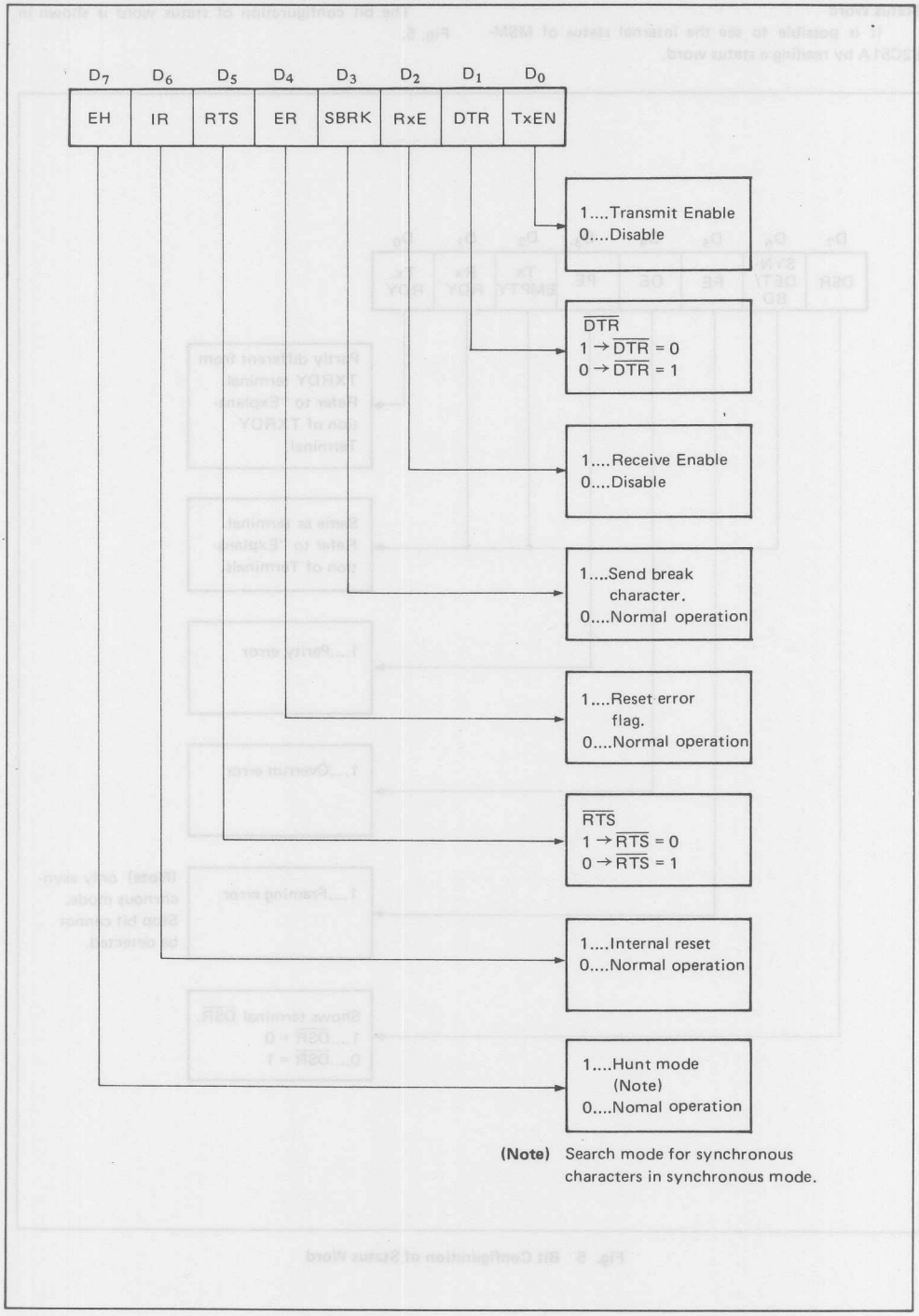


Fig. 4 Bit Configuration of Command

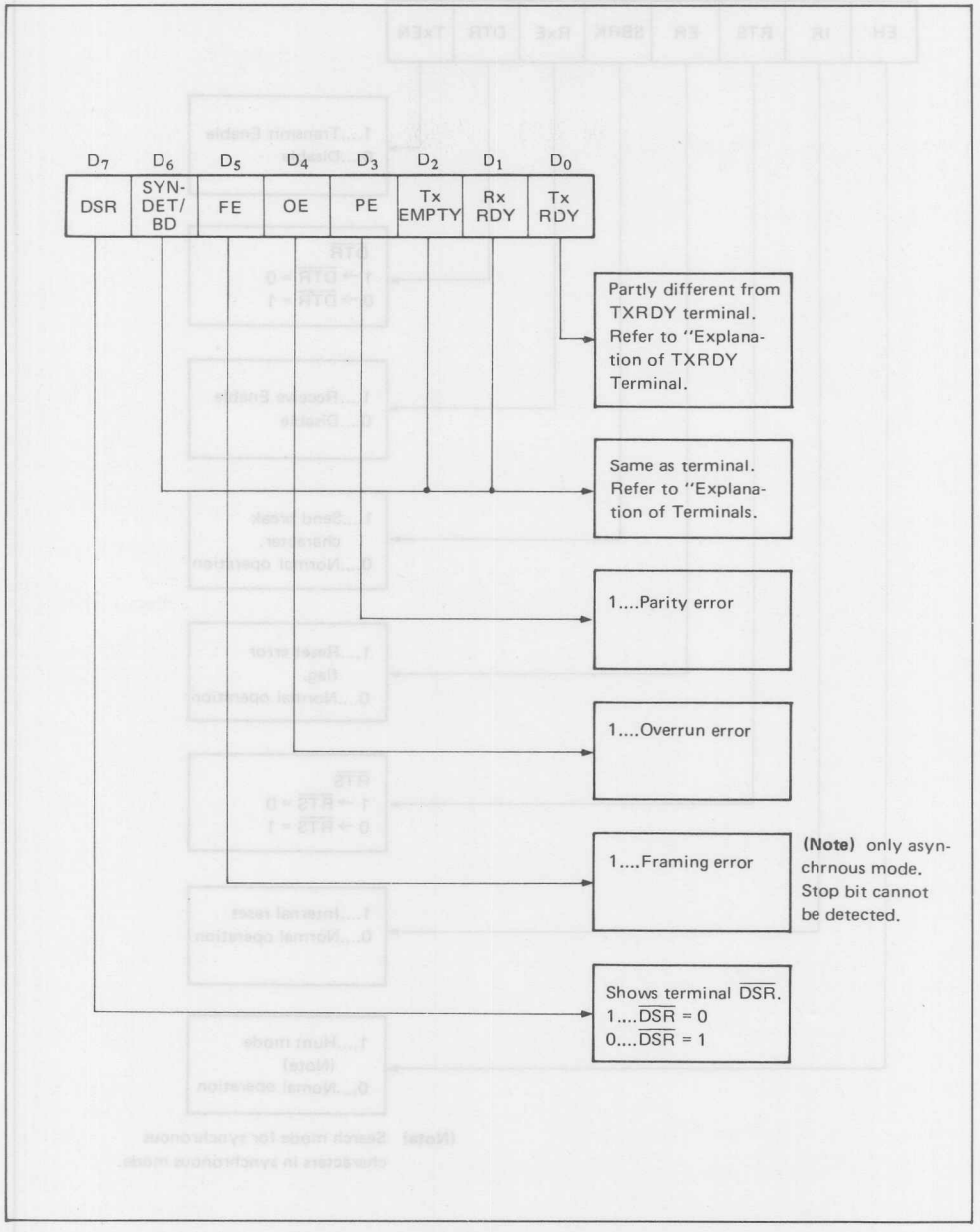


Fig. 5 Bit Configuration of Status Word

Standby Status

It is possible to put MSM82C51A in "standby status" for the complete static configuration of CMOS.

It is when the following conditions have been satisfied that MSM82C51A is in "standby status."

(1) $\overline{CS}$ terminal shall be fixed at Vcc level.

(2) Input pins other than $\overline{CS}$, D_0 to D_7, $\overline{RD}$, $\overline{WR}$ and C/$\overline{D}$ shall be fixed at Vcc or GND level (including SYNDET in external synchronous mode).

Note When all outputs current are 0, ICCS specification is applied.

Explanation of Each Terminal

D_0 to D_7 (I/O terminal)

This is a bidirectional data bus which receive control word and transmit data from CPU and send status word and received data to CPU.

RESET (Input terminal)

A "High" on this input forces the MSM82C51A into "reset status."

The device waits for the writing of "mode instruction."

The min. reset width is six clock inputs during the operating status of CLK.

CLK (Input terminal)

CLK signal is used to generate an internal device timing.

CLK signal is independent of $\overline{RXC}$ or $\overline{TXC}$.

However, the frequency of CLK must be greater than 30 times the $\overline{RXC}$ and $\overline{TXC}$ at Synchronous mode and Asynchronous "x1" mode, and must be greater than 5 times at Asynchronous "x16" and "x64" mode.

$\overline{WR}$ (Input terminal)

This is "active low" input terminal which receives a signal for writing transmit data and control words from CPU into MSM82C51A.

$\overline{RD}$ (Input terminal)

This is "active low" input terminal which receives a signal for reading receive data and status words from MSM82C51A.

C/$\overline{D}$ (Input terminal)

This is an input terminal which receives a signal for selecting data or command word and status word when MSM82C51A is accessed by CPU.

If C/$\overline{D}$ = low, data will be accessed.

If C/$\overline{D}$ = high, command word or status word will be accessed.

$\overline{CS}$ (Input terminal)

This is "active low" input terminal which selects the MSM82C51A at low level when CPU accesses.

Note The device won't be in "standby status" only setting $\overline{CS}$ = High.
Refer to "Explanation of Standby Status."

TXD (Output terminal)

This is an output terminal for transmit data from which serial-converted data is sent out.

The device is in "mark status" (high level) after resetting or during a status when transmit is disable.

It is also possible to set the device in "break status" (low level) by a command.

TXRDY (Output terminal)

This is an output terminal which indicate that MSM82C51A is ready to accept a transmit data character. But the terminal is always at low level if $\overline{CTS}$ = high or the device was set in "TX disable status" by a command.

Note TXRDY of status word indicates that transmit data character is receivable, regardless of $\overline{CTS}$ or command.

If CPU write a data character, TXRDY will be reset by the leadingedge or $\overline{WR}$ signal.

TXEMPTY (Output terminal)

This is an output terminal which indicates that MSM82C51A transmitted all the characters and had no data character.

In "synchronous mode," the terminal is at high level, if transmit data characters are no longer left and sync characters are automatically transmitted.

If CPU write a data character, TXEMPTY will be reset by the leadingedge of $\overline{WR}$ signal.

Note As transmitter is disabled by setting CTS "High" or command, a data written before disabled will be sent out, then TXD and TXEMPTY will be "High".
Even if a data is written after disabled, that data is not sent out and TXE will be "High".
After enabled transmitter, it sent out.
(Refer to Timing Chart of Transmitter Control and Flag Timing)

$\overline{TXC}$ (Input terminal)

This is a clock input signal which determines the transfer speed of transmit data.

In "synchronous mode," the baud rate will be the same as the frequency of $\overline{TXC}$.

In "Asynchronous mode", it is possible to select baud rate factor by mode instruction.

It can be 1, 1/16 or 1/64 the $\overline{TXC}$.

The falling edge of $\overline{TXC}$ sifts the serial data out of the MSM82C51A.

RXD (Input terminal)

This is a terminal which receives serial data.

RXRDY (Output terminal)

This is a terminal which indicates that MSM82C51A contains a character that is ready to READ.

If CPU read a data character, RXRDY will be reset by the leadingedge of $\overline{RD}$ signal.

Unless CPU reads a data character before next one character is received completely, the preceding data will be lost. In such a case, an overrun error flag of status word will be set.

$\overline{RXC}$ (Input terminal)

This is a clock input signal which determines the transfer speed of receive data.

In "synchronous mode," the baud rate will be the same as the frequency of $\overline{RXC}$.

In "asynchronous mode," it is possible to select baud rate factor by mode instruction.

It can be 1, 1/16, 1/64 the $\overline{RXC}$.

SYNDET/BD (Input or output terminal)

This is a terminal which function changes according to mode.

In "internal synchronous mode," this terminal is at high level, if sync characters are received and synchronized. If status word is read, the terminal will be reset.

In "external synchronous mode," this is an input terminal.

If "High" on this input forces, MSM82C51A starts receiving data character.

In "asynchronous mode," this is an output terminal which generates "high level" output upon the detection of "break" character, if receiver data contained "low-level" space between stop bits of two continuous characters. The terminal will be reset, if RXD is at high level.

$\overline{DSR}$ (Input terminal)

This is an input port for MODEM interface. The input status of the terminal can be recognized by CPU reading status words.

$\overline{DTR}$ (Output terminal)

This is an output port for MODEM interface. It is possible to set the status of $\overline{DTR}$ by a command.

$\overline{CTS}$ (Input terminal)

This is an input terminal for MODEM interface which is used for controlling a transmit circuit. The terminal controls data transmit if the device is set in "TX Enable" status by a command. Data is transmitable if the terminal is at low level.

$\overline{RTS}$ (Output terminal)

This is an output port for MODEM interface. It is possible to set the status of $\overline{RTS}$ by a command.

ABSOLUTE MAXIMUM RATINGS

Parameter	Symbol	Limits		Unit	Conditions
		MSM82C51ARS	MSM82C51AGS		
Power supply voltage	V_{CC}	−0.5 ~ +7		V	With respect to GND
Input voltage	V_{IN}	−0.5 ~ V_{CC} + 0.5		V	
Output voltage	V_{OUT}	−0.5 ~ V_{CC} + 0.5		V	
Storage temperature	T_{stg}	−55 ~ 150		°C	—
Power dissipation	P_D	0.9	0.7	W	Ta = 25°C

OPERATING RANGE

Parameter	Symbol	Limits	Unit
Power supply voltage	V_{CC}	3 ~ 6	V
Operating temperature	T_{OP}	−40 ~ 85	°C

RECOMMENDED OPERATING CONDITIONS

Parameter	Symbol	Min.	Typ.	Max.	Unit
Power supply voltage	V_{CC}	4.5	5	5.5	V
Operating temperature	T_{OP}	−40	+25	+85	°C
"L" input voltage	V_{IL}	−0.3		+0.8	V
"H" input voltage	V_{IH}	2.2		V_{CC} + 0.3	V

DC CHARACTERISTICS

(V_{CC} = 4.5 ~ 5.5V Ta = −40°C + 85°C)

Parameter	Symbol	Min.	Typ.	Max.	Unit	Measurement Conditions
"L" output voltage	V_{OL}			0.45	V	I_{OL} = 2 mA
"H" output voltage	V_{OH}	3.7			V	I_{OH} = −400 μA
Input leak current	I_{LI}	−10		10	μA	$0 \leq V_{IN} \leq V_{CC}$
Output leak current	I_{LO}	−10		10	μA	$0 \leq V_{OUT} \leq V_{CC}$
Operating supply current	I_{CCO}			5	mA	Asynchronous X64 during transmitting/receiving
Standby supply current	I_{CCS}			100	μA	All input voltage shall be fixed at V_{CC} or GND level.

CPU Bus Interface Part

Parameter	Symbol	Min.	Max.	Unit	Remarks
Address stable before $\overline{RD}$	t_{AR}	20		NS	Note 2
Address hold time for $\overline{RD}$	t_{RA}	20		NS	Note 2
$\overline{RD}$ pulse width	t_{RR}	250		NS	
Data delay from $\overline{RD}$	t_{RD}		200	NS	
$\overline{RD}$ to data float	t_{DF}	10	100	NS	
Address stable before $\overline{WR}$	t_{AW}	20		NS	Note 2
Address hold time for $\overline{WR}$	t_{WA}	20		NS	Note 2
$\overline{WR}$ pulse width	t_{WW}	250		NS	
Data set-up time for $\overline{WR}$	t_{DW}	150		NS	
Data hold time for $\overline{WR}$	t_{WD}	20		NS	
Recovery time between $\overline{WR}$	t_{RV}	6		T_{cy}	
RESET pulse width	t_{RESW}	6		T_{cy}	

Serial Interface Part

Parameter		Symbol	Min.	Max.	Unit	Remarks
Main clock period		t_{cy}	250		NS	Note 3
Clock low time		t_{ϕ}	90		NS	
Clock high time		t_{ϕ}	120	$t_{cy}-90$	NS	
Clock rise/fall time		t_R, t_F		20	NS	
TXD delay from falling edge of $\overline{TXC}$		t_{DTX}		1	μS	
Transmitter clock frequency	1X Baud	f_{TX}	DC	64	kHz	Note 3
	16X, Baud	f_{TX}	DC	615	kHz	
	64X, Baud	f_{TX}	DC	615	kHz	
Transmitter clock low time	1X Baud	t_{TPW}	13		T_{cy}	
	16X, 64X Baud	t_{TPW}	2		T_{cy}	
Receiver clock high time	1X Baud	t_{TPD}	15		T_{cy}	
	16X, 64X Baud	t_{TPD}	3		T_{cy}	
Receiver clock frequency	1X Baud	f_{RX}	DC	64	kHz	Note 3
	16X Baud	f_{RX}	DC	615	kHz	
	64X Baud	f_{RX}	DC	615	kHz	
Receiver clock low time	1X Baud	t_{RPW}	13		T_{cy}	
	16X, 64X Baud	t_{RPW}	2		T_{cy}	
Receiver clock high time	1X Baud	t_{RPD}	15		T_{cy}	
	16X, 64X Baud	t_{RPD}	3		T_{cy}	
Time from the center of last bit to the rise of RXRDY		t_{TXRDY}		8	T_{cy}	
Time from the leading edge of $\overline{WR}$ to the fall of RXRDY		$t_{TXRDY\ CLEAR}$		400	NS	
Time from the center of last bit to the rise of RXRDY		t_{RXRDY}		26	T_{cy}	

Parameter	Symbol	Min.	Max.	Unit	Remarks
Time from the leading edge of $\overline{RD}$ to the fall of RXRDY	$t_{RXRDY\ CLEAR}$		400	NS	
Internal SYNDET delay time from rising edge of $\overline{RXC}$	t_{IS}		26	T_{cy}	
SYNDET setup time for $\overline{RXC}$	t_{ES}	18		T_{cy}	
TXE delay time from the center of last bit	$t_{TXEMPTY}$	20		T_{cy}	
MODEM control signal delay time from rising edge of $\overline{WR}$	t_{WC}	8		T_{cy}	
MODEM control signal setup time for falling edge of $\overline{RD}$	t_{CR}	20		T_{cy}	
RXD setup time for rising edge of $\overline{RXC}$ (1X Baud)	t_{RXDS}	11		T_{cy}	
RXD hold time for falling edge of $\overline{RXC}$ (1X Baud)	t_{RXDH}	17		T_{cy}	

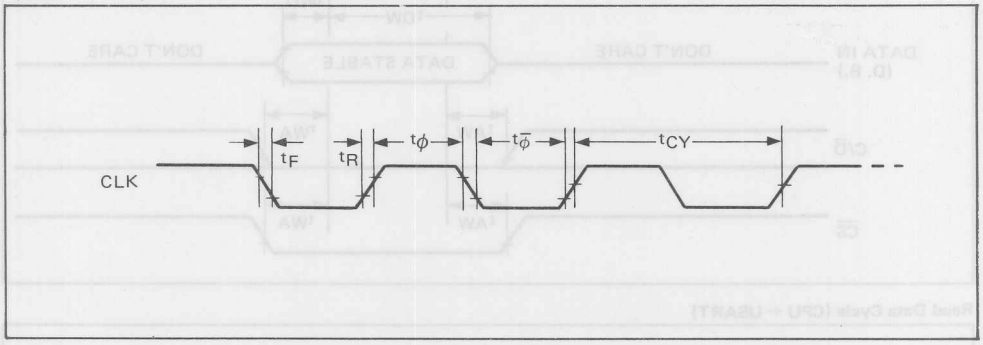
Caution 1) AC characteristics are measured at 150 pF capacity load as an output load based on 0.8 V at low level and 2.2 V at high level for output and 1.5 V for input.

2) Addresses are $\overline{CS}$ and $C/\overline{D}$.

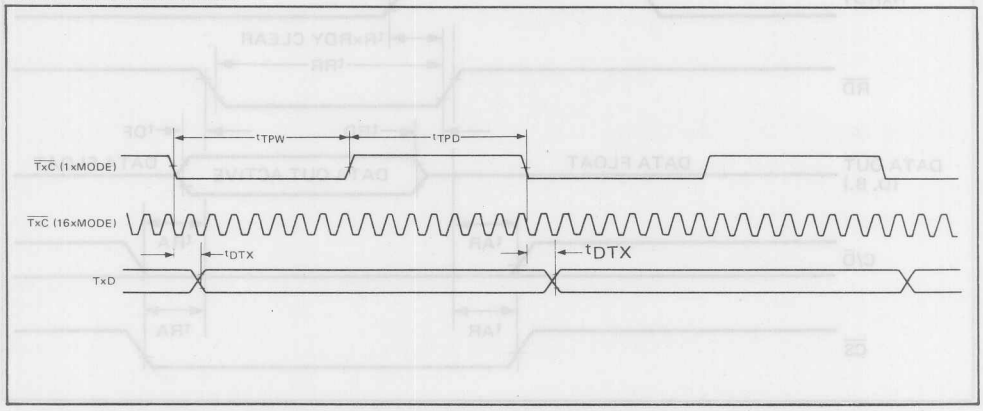
3) f_{TX} or $f_{RX} \leq 1/(30\ Tcy)$ 1 x baud
f_{TX} or $f_{RX} \leq 1/(5\ Tcy)$ 16 x, 64 x Baud

TIMING CHART

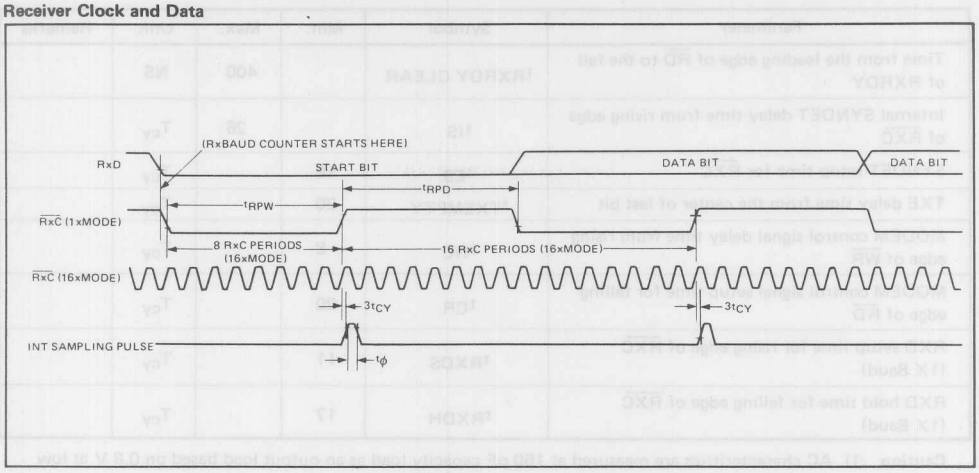
System Clock Input



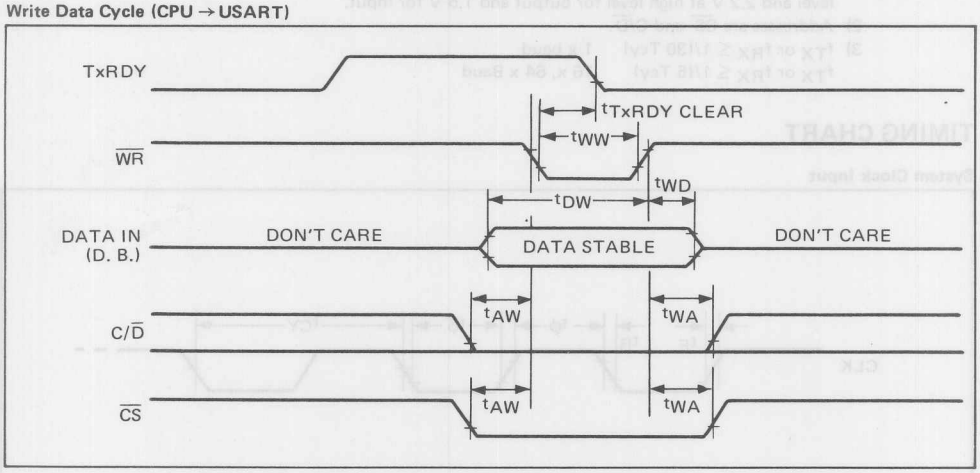
Transmitter Clock and Data



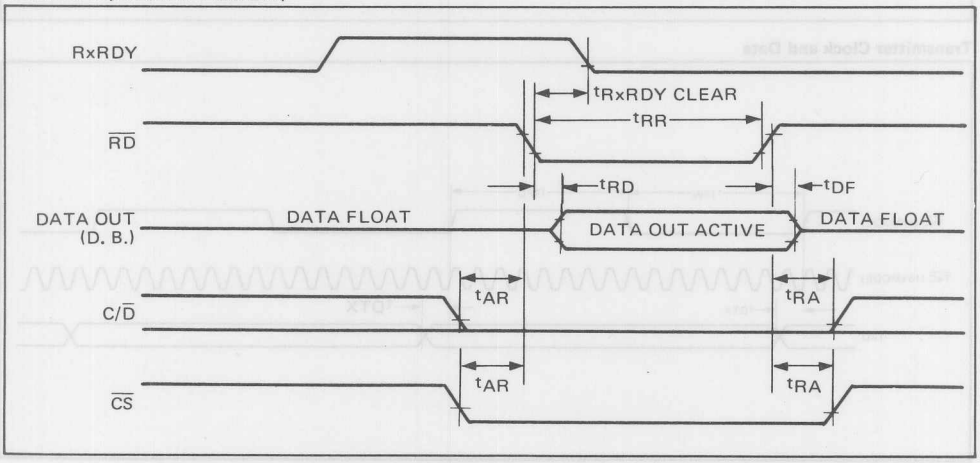
Receiver Clock and Data



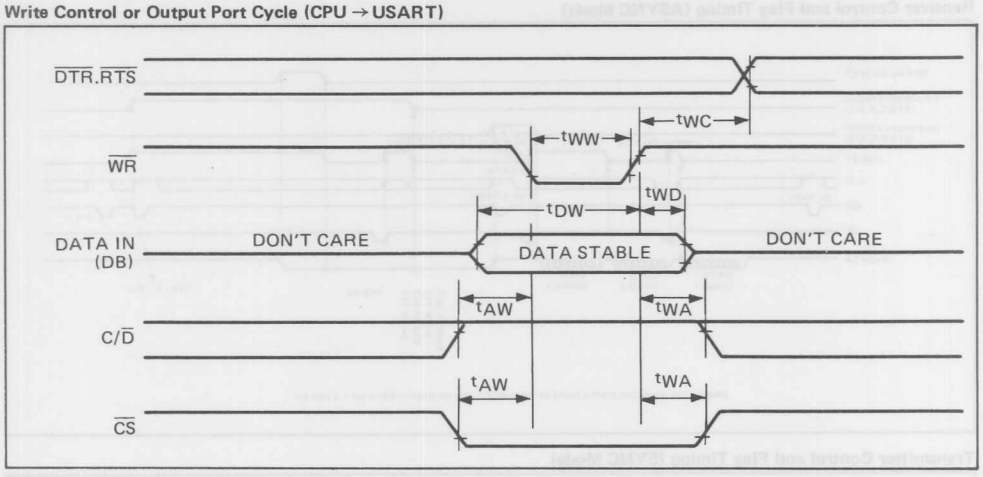
Write Data Cycle (CPU → USART)



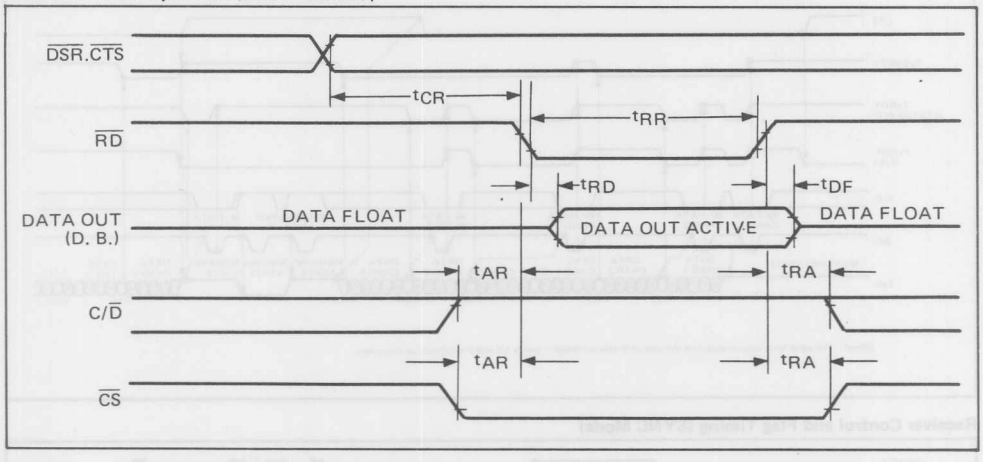
Read Data Cycle (CPU ← USART)



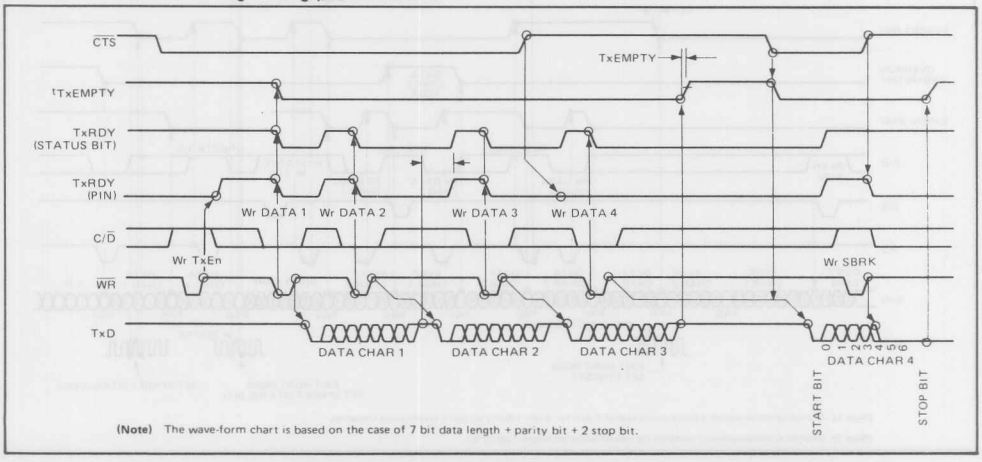
Write Control or Output Port Cycle (CPU → USART)



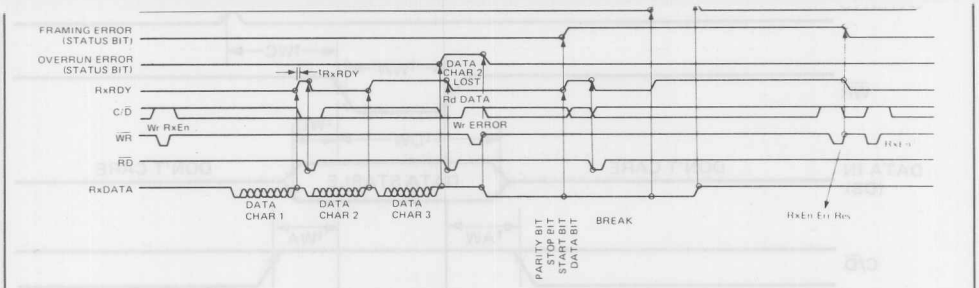
Read Control or Input Port (CPU ← USART)



Transmitter Control and Flag Timing (ASYNC Mode)

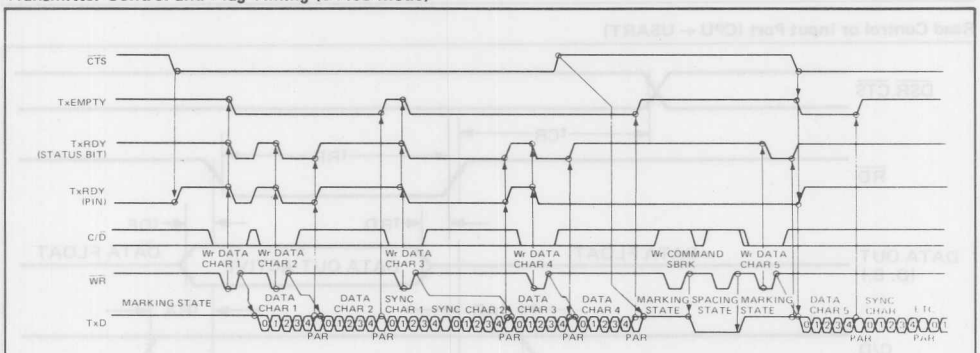


(Note) The wave-form chart is based on the case of 7 bit data length + parity bit + 2 stop bit.



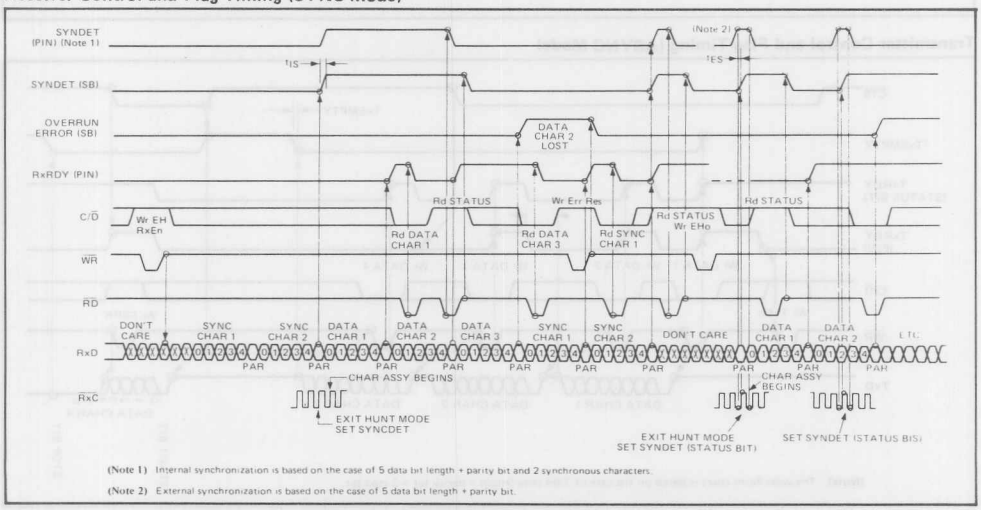
(Note) The wave-form chart is based on the case of 7 data bit length + parity bit + 2 stop bit.

Transmitter Control and Flag Timing (SYNC Mode)



(Note) The wave-form chart is based on the case of 5 data bit length + parity bit and 2 synchronous characters.

Receiver Control and Flag Timing (SYNC Mode)



(Note 1) Internal synchronization is based on the case of 5 data bit length + parity bit and 2 synchronous characters.

(Note 2) External synchronization is based on the case of 5 data bit length + parity bit.

OKI semiconductor

MSM82C53-5RS/GS

CMOS PROGRAMMABLE INTERVAL TIMER

GENERAL DESCRIPTION

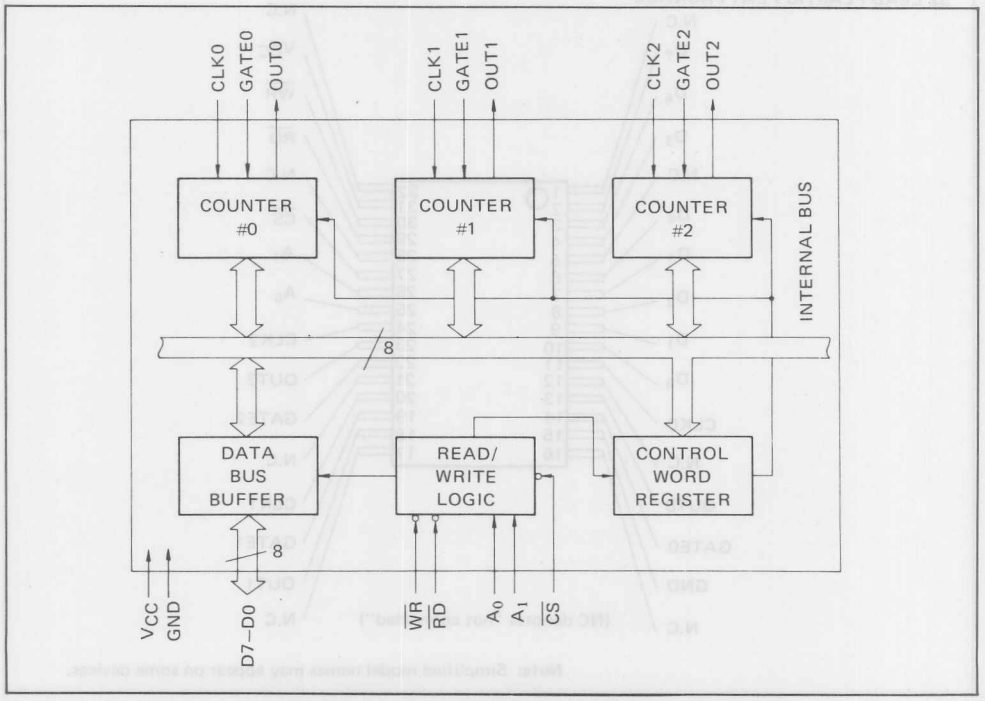
The MSM82C53-5RS/GS is a programmable universal timer designed for use in microcomputer systems. Based on silicon gate CMOS technology, it requires a standby current of only 100 μA (max.) when the chip is in the non-selected state. And during timer operation, the power consumption is still very low with only 5 mA (max.) of current required.

It consists of three independent counters, and can count up to a maximum of 5 MHz. The timer features six different counter modes, and binary count/BCD count functions. Count values can be set in byte or word units, and all functions are freely programmable.

FEATURES

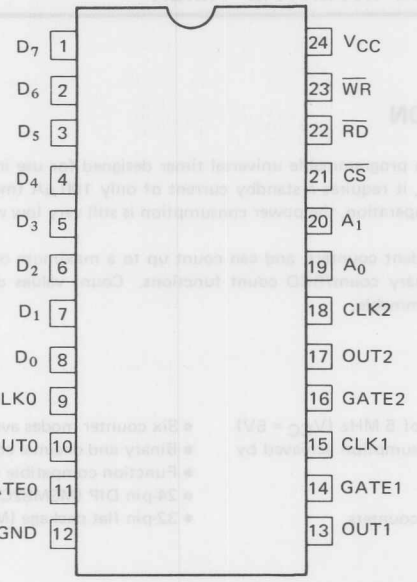
- Maximum operating frequency of 5 MHz (V_{CC} = 5V)
- High speed and low power consumption achieved by silicon gate CMOS technology.
- Completely static operation
- Three independent 16-bit down-counters
- 3V to 6V single power supply
- Six counter modes available for each counter
- Binary and decimal counting possible
- Function compatible with 8253-5
- 24-pin DIP (MSM82C53-5RS)
- 32-pin flat package (MSM82C53-5GS)

FUNCTIONAL BLOCK DIAGRAM

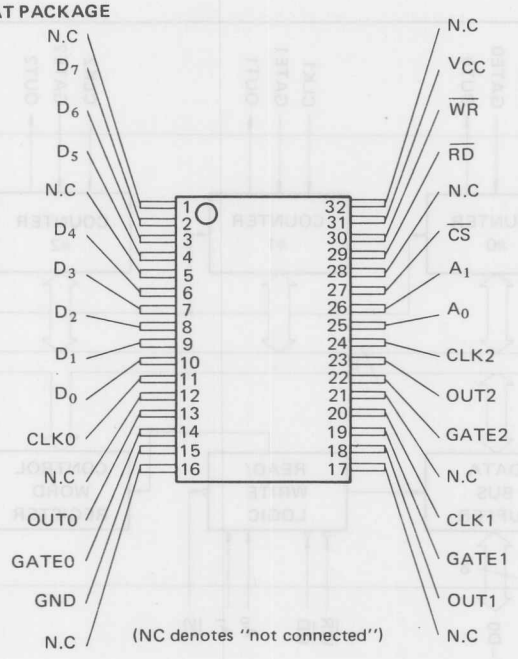


PIN CONFIGURATION

MSM82C53-5RS (TOP VIEW)
24 LEAD PLASTIC DIP



MSM82C53-5GS (TOP VIEW)
32 LEAD PLASTIC FLAT PACKAGE



(NC denotes "not connected")

Note: Simplified model names may appear on some devices.

ABSOLUTE MAXIMUM RATINGS

Parameter	Symbol	Conditions	Limits		Unit
			MSM82C53-5RS	MSM82C53-5GS	
Supply Voltage	V_{CC}	Respect to GND	−0.5 to +7		V
Input Voltage	V_{IN}		−0.5 to V_{CC} + 0.5		V
Output Voltage	V_{OUT}		−0.5 to V_{CC} + 0.5		V
Storage Temperature	Tstg		−55 to + 150		°C
Permissible Loss	P_D	Ta = 25°C	0.9	0.7	W

OPERATIONAL RANGES

Parameter	Symbol	Limits	Conditions	Unit
Supply Voltage	V_{CC}	3 to 6	V_{IL} = 0.2V, V_{IH} = V_{CC} − 0.2V, operating frequency 2.6 MHz	V
Operating Temperature	T_{OP}	−40 to +85		°C

RECOMMENDED OPERATING CONDITIONS

Parameter	Symbol	Min.	Typ.	Max.	Unit
Supply Voltage	V_{CC}	4.5	5	5.5	V
Operating Temperature	T_{OP}	−40	+25	+85	°C
"L" Input Voltage	V_{IL}	−0.3		+0.8	V
"H" Input Voltage	V_{IH}	2.2		V_{CC} + 0.3	V

DC CHARACTERISTICS

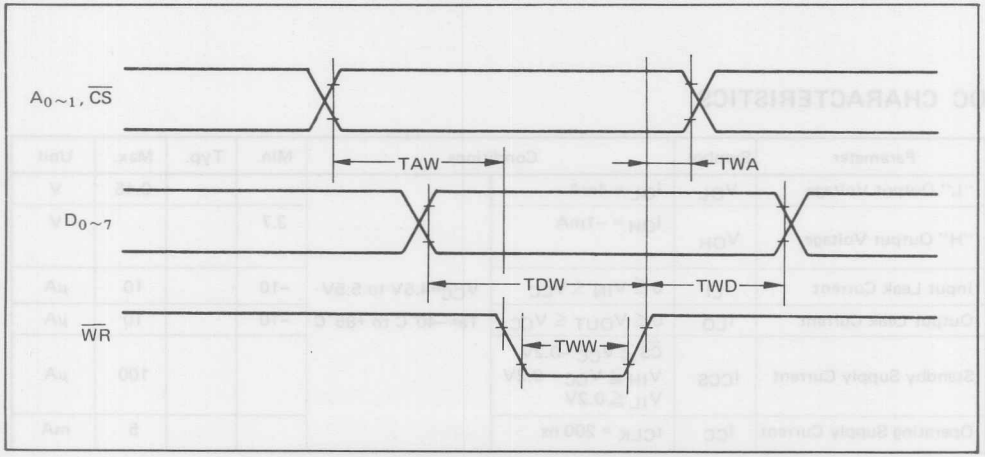
Parameter	Symbol	Conditions		Min.	Typ.	Max.	Unit
"L" Output Voltage	V_{OL}	I_{OL} = 4mA	V_{CC}=4.5V to 5.5V Ta=−40°C to +85°C			0.45	V
"H" Output Voltage	V_{OH}	I_{OH} = −1mA		3.7			V
Input Leak Current	I_{LI}	$0 \leq V_{IN} \leq V_{CC}$		−10		10	μA
Output Leak Current	I_{LO}	$0 \leq V_{OUT} \leq V_{CC}$		−10		10	μA
Standby Supply Current	I_{CCS}	$\overline{CS} \geqq V_{CC}$ −0.2V $V_{IH} \geqq V_{CC}$ − 0.2V $V_{IL} \leqq 0.2V$				100	μA
Operating Supply Current	I_{CC}	t_{CLK} = 200 ns				5	mA

Parameter	Symbol	Min.	Max.	Unit	Conditions	
Address Set-up Time before reading	TAR	30		ns		C_L = 150pF
Address Hold Time after reading	TRA	0		ns	Read cycle	
Read Pulse Width	TRR	150		ns		
Read Recovery Time	TRVR	200		ns		
Address Set-up Time before writing	TAW	0		ns		
Address Hold Time after writing	TWA	30		ns		
Write Pulse Width	TWW	150		ns	Write cycle	
Data Input Set-up Time before writing	TDW	100		ns		
Data Input Hold Time after writing	TWD	30		ns		
Write Recovery Time	TRVW	200		ns		
Clock Cycle Time	TCLK	200	D.C.	ns		
Clock "H" Pulse Width	TPWH	60		ns		
Clock "L" Pulse Width	TPWL	60		ns	Clock and gate timing	
"H" Gate Pulse Width	TGW	50		ns		
"L" Gate Pulse Width	TGL	50		ns		
Gate Input Set-up Time before clock	TGS	50		ns		
Gate Input Hold Time after clock	TGH	50		ns		
Output Delay Time after reading	TRD		120	ns		
Output Floating Delay Time after reading	TDF	5	90	ns		
Output Delay Time after gate	TODG		120	ns	Delay time	
Output Delay Time after clock	TOD		150	ns		
Output Delay Time after address	TAD		180	ns		

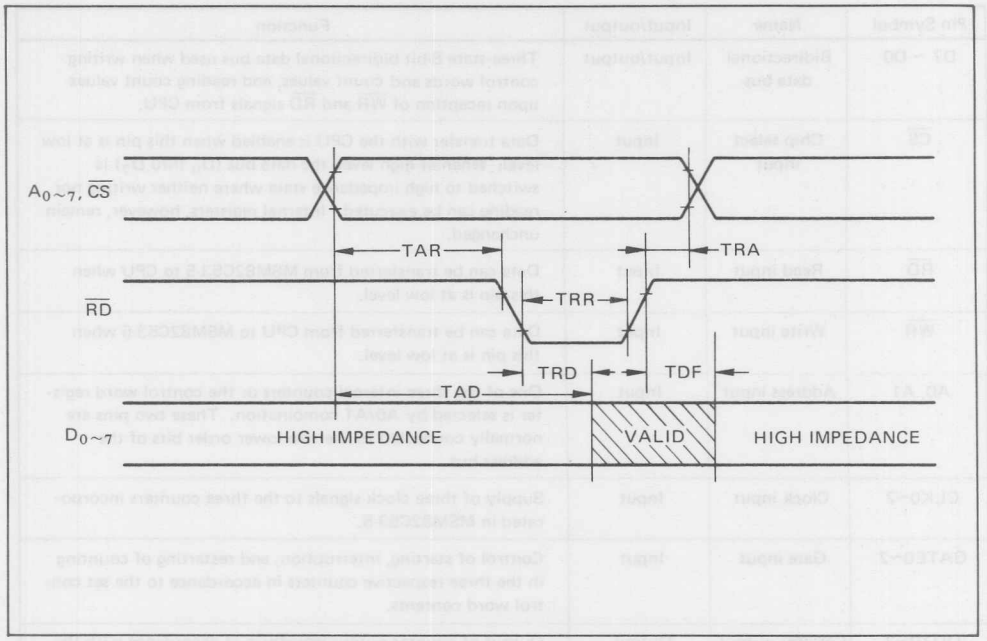
Note: Timing measured at V_L = 0.8V and V_H = 2.2V for both inputs and outputs.

TIME CHART

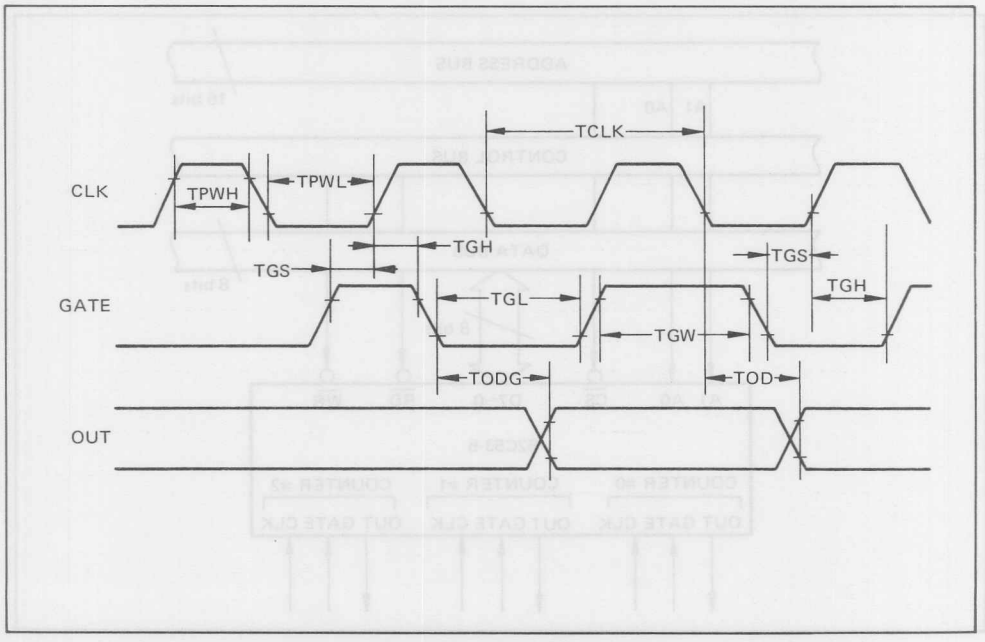
Write Timing



Read Timing



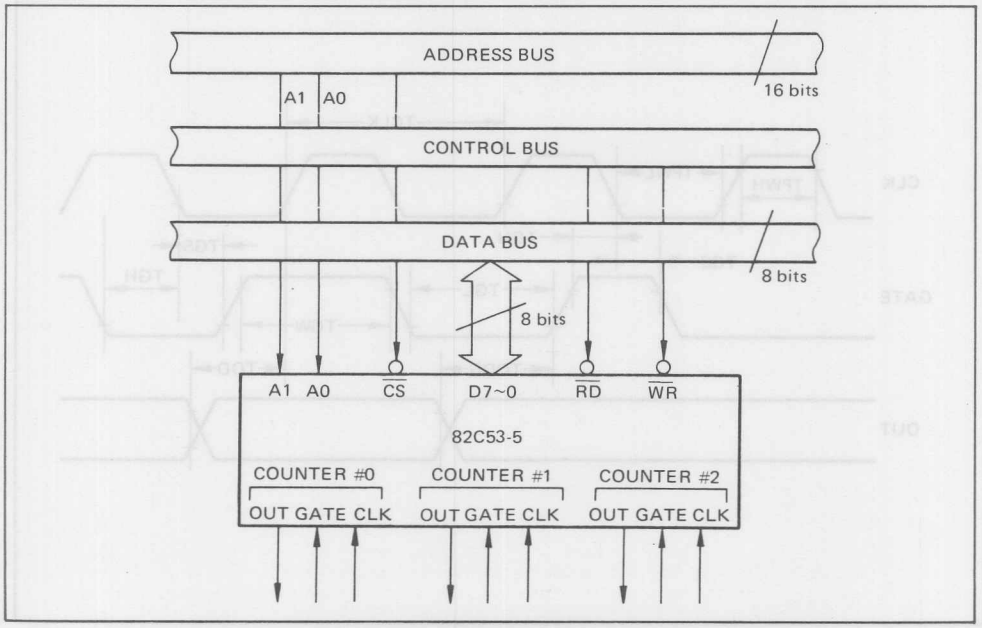
Clock & Gate Timing



DESCRIPTION OF PIN FUNCTIONS

Pin Symbol	Name	Input/output	Function
D7 ~ D0	Bidirectional data bus	Input/output	Three-state 8-bit bidirectional data bus used when writing control words and count values, and reading count values upon reception of $\overline{WR}$ and $\overline{RD}$ signals from CPU.
$\overline{CS}$	Chip select input	Input	Data transfer with the CPU is enabled when this pin is at low level. When at high level, the data bus (D_0 thru D_7) is switched to high impedance state where neither writing nor reading can be executed. Internal registers, however, remain unchanged.
$\overline{RD}$	Read input	Input	Data can be transferred from MSM82C53-5 to CPU when this pin is at low level.
$\overline{WR}$	Write input	Input	Data can be transferred from CPU to MSM82C53-5 when this pin is at low level.
A0, A1	Address input	Input	One of the three internal counters or the control word register is selected by A0/A1 combination. These two pins are normally connected to the two lower order bits of the address bus.
CLK0~2	Clock input	Input	Supply of three clock signals to the three counters incorporated in MSM82C53-5.
GATE0~2	Gate input	Input	Control of starting, interruption, and restarting of counting in the three respective counters in accordance to the set control word contents.
OUT0~2	Counter output	Output	Output of counter output waveform in accordance with the set mode and count value.

SYSTEM INTERFACING



DESCRIPTION OF BASIC OPERATIONS

Data transfers between the internal registers and the external data bus is outlined in the following table.

$\overline{CS}$	$\overline{RD}$	$\overline{WR}$	A1	A0	Function
0	1	0	0	0	Data bus to counter # 0 Writing
0	1	0	0	1	Data bus to counter # 1 Writing
0	1	0	1	0	Data bus to counter # 2 Writing
0	1	0	1	1	Data bus to control word register Writing
0	0	1	0	0	Data bus from counter # 0 Reading
0	0	1	0	1	Data bus from counter # 1 Reading
0	0	1	1	0	Data bus from counter # 2 Reading
0	0	1	1	1	Data bus in high impedance status
1	x	x	x	x	
0	1	1	x	x	

x denotes "not specified".

DESCRIPTION OF OPERATION

82C53-5 functions are selected by control word from CPU. In the required program sequence, the control word setting is followed by the count value setting and execution of the desired timer operation.

Control Word and Count Value Program

Each counter operating mode is set by control word programming. The control word format is outlined below.

D7	D6	D5	D4	D3	D2	D1	D0
SC1	SC0	RL0	RL1	M2	M1	M0	BCD
Select Counter		Read/Load		Mode			BCD

($\overline{CS}$ = 0, A0, A1 = 1,1, $\overline{RD}$ = 1, $\overline{WR}$ = 0)

- **Select Counter (SC0, SC1):** Selection of set counter

SC1	SC0	Set Contents
0	0	Counter # 0 selection
0	1	Counter # 1 selection
1	0	Counter # 2 selection
1	1	Illegal combination

- **Read/Load (RL1, RL0):** Count value Reading/Loading format setting

RL1	RL0	Set Contents
0	0	Counter Latch operation
0	1	Reading/Loading of Least Significant byte (LSB)
1	0	Reading/Loading of Most Significant byte (MSB)
1	1	Reading/Loading of LSB followed by MSB

- **Mode (M2, M1, M0):** Operation waveform mode setting

M2	M1	M0	Set Contents
0	0	0	Mode 0 (Interrupt on Terminal Count)
0	0	1	Mode 1 (Programmable One-Shot)
x	1	0	Mode 2 (Rate Generator)
x	1	1	Mode 3 (Square Wave Generator)
1	0	0	Mode 4 (Software Triggered Strobe)
1	0	1	Mode 5 (Hardware Triggered Strobe)

x denotes "not specified".

- **BCD:** Operation count mode setting

BCD	Set Contents
0	Binary Count (16-bits Binary)
1	BCD Count (4-decades Binary Coded Decimal)

After setting Read/Load, Mode, and BCD in each counter as outlined above, next set the desired count value. (In some Modes, counting is started immediately after the count value has been written). This count value setting must conform with the Read/Load format set in advance. And note that the internal counters are reset to 0000H during control word setting.

If the two bytes (LSB and MSB) are written at this stage (RL0 and RL1 = 1,1), take note of the following precaution.

Although the count values may be set in the three counters in any sequence after the control word has been set in each counter, count values must be set consecutively in the LSB — MSB order in any one counter.

- **Example of control word and count value setting**

Counter # 0: Read/Load LSB only, Mode 3, Binary count, count value 3H
Counter # 1: Read/Load MSB only, Mode 5, Binary count, count value AA00H
Counter # 2: Read/Load LSB and MSB, Mode 0, BCD count, count value 1234

```
MVI A, 1EH
OUT n3          Counter #0 control word setting
MVI A, 6AH
OUT n3          Counter #1 control word setting
MVI A, B1H
OUT n3          Counter #2 control word setting
MVI A, 03H
OUT n0          Counter #0 count value setting
MVI A, AAH
OUT n1          Counter #1 count value setting
MVI A, 34H
OUT n2          Counter #2 count value setting
MVI A, 12H      (LSB then MSB)
OUT n2
```

Note: n0: Counter #0 address
n1: Counter #1 address
n2: Counter #2 address
n3: Control word register address

- **The minimum and maximum count values which can be counted in each mode are listed below.**

Mode	Min.	Max.	Remarks
0	1	0	0 executes 10000H count (ditto in other modes)
1	1	0	
2	2	0	1 cannot be counted
3	2	1	1 executes 10001H count
4	1	0	
5	1	0	

Mode Definition

- **Mode 0 (terminal count)**

The counter output is set to "L" level by the mode setting. If the count value is then written in the counter with the gate input at "H" level (that is, upon completion of writing the MSB when there are two bytes), the clock input counting is started. When the terminal count is reached, the output is switched to "H" level and is maintained in this status until the control word and count value are set again.

Counting is interrupted if the gate input is switched to "L" level, and restarted when switched back to "H" level.

When Count Values are written during counting, the operation is following.

1 byte Read/Load.... When the new count value is written, counting is stopped immediately, and then restarted at the new count value by the next clock.

2-byte Read/Load.... When byte 1 (LSB) of the new count value is written, counting is stopped immediately. Counting is restarted at the new count value when byte 2 (MSB) is written.

- **Mode 1 (programmable one-shot)**

The counter output is switched to "H" level by the mode setting. Note that in this mode, counting is not started if only the count value is written. Since counting has to be started in this mode by using the leading edge of the gate input as a trigger, the counter output is switched to "L" level by the next clock after the gate input trigger. This "L" level status is maintained during the set count value, and is switched back to "H" level when the terminal count is reached.

Once counting has been started, there is no interruption until the terminal count is reached, even if the gate input is switched to "L" level in the meantime. And although counting continues even if a new count value is written during the counting, counting is started at the new count value if another trigger is applied by the gate input.

- **Mode 2 (rate generator)**

The counter output is switched to "H" level by the mode setting. When the gate input is at "H" level, counting is started by the next clock after the count value has been written. And if the gate input is at "L" level, counting is started by using the rising edge of the gate input as a trigger after the count value has been set.

An "L" level output pulse appears at the counter output during a single clock duration once every n clock inputs where n is the set count value. If a new count value is written during while counting is in progress, counting is started at the new count value following output of the pulse currently being counted. And if the gate input is switched to "L" level during counting, the counter output is forced to switch to "H" level, the counting being restarted by the rising edge of the gate input.

- **Mode 3 (square waveform rate generator)**

The counter output is switched to "H" level by the mode setting. Counting is started in the same way as described for mode 2 above.

The repeated square wave output appearing at the counter output contains half the number of counts as the set count value. If the set count value (n) is an odd number, the repeated square wave output consists of only $(n + 1)/2$ clock inputs at "H" level and $(n - 1)/2$ clock inputs at "L" level.

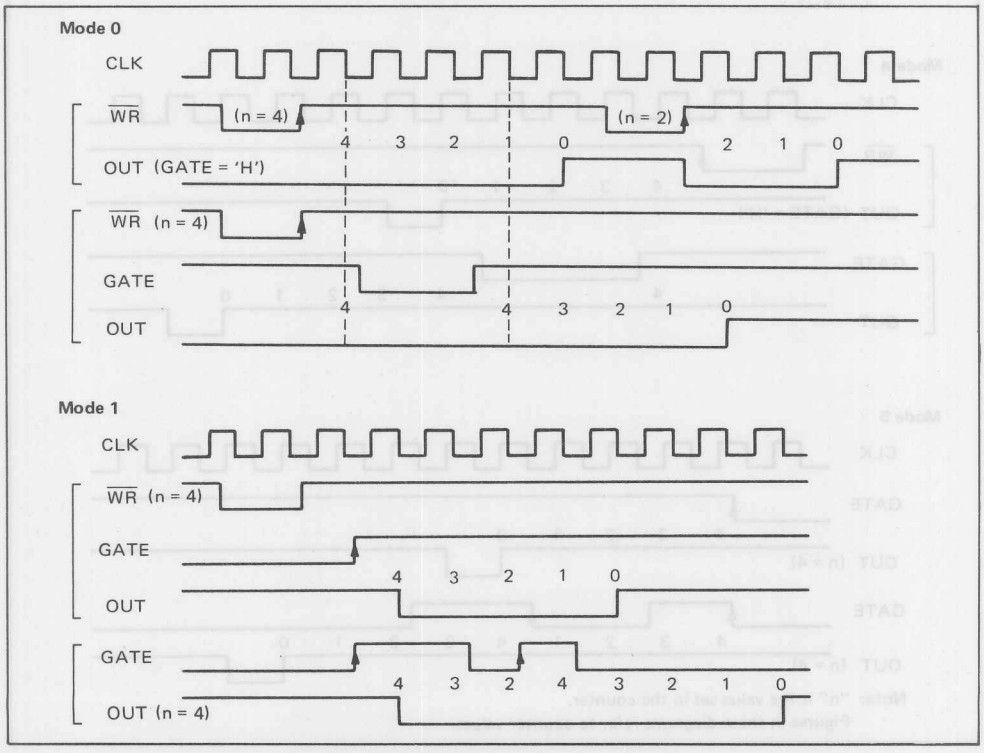
If a new count value is written during counting, the new count value is reflected immediately after the

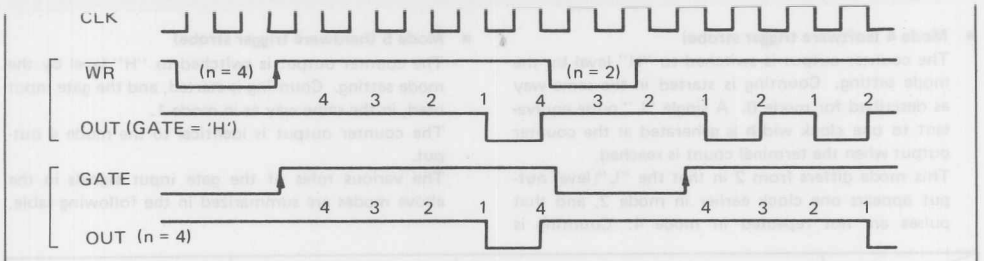
change ("H" to "L" or "L" to "H") in the next counter output to be executed. The counting operation at the gate input is done the same as in mode 2.

- **Mode 4 (software trigger strobe)**
  The counter output is switched to "H" level by the mode setting. Counting is started in the same way as described for mode 0. A single "L" pulse equivalent to one clock width is generated at the counter output when the terminal count is reached.
  This mode differs from 2 in that the "L" level output appears one clock earlier in mode 2, and that pulses are not repeated in mode 4. Counting is stopped when the gate input is switched to "L" level, and restarted from the set count value when switched back to "H" level.

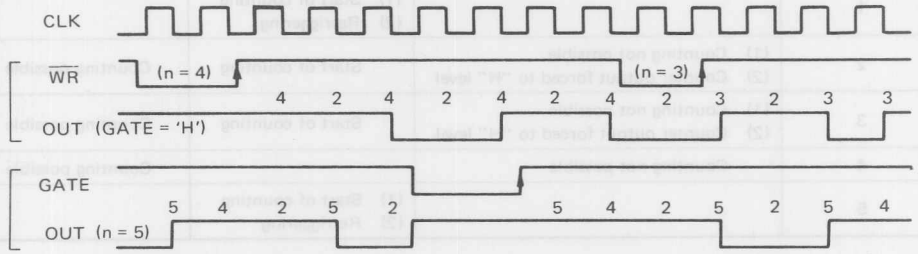
- **Mode 5 (hardware trigger strobe)**
  The counter output is switched to "H" level by the mode setting. Counting is started, and the gate input used, in the same way as in mode 1.
  The counter output is identical to the mode 4 output.
  The various roles of the gate input signals in the above modes are summarized in the following table.

Mode \ Gate	"L" Level Falling Edge	Rising Edge	"H" Level
0	Counting not possible		Counting possible
1		(1) Start of counting (2) Retriggering	
2	(1) Counting not possible (2) Counter output forced to "H" level	Start of counting	Counting possible
3	(1) Counting not possible (2) Counter output forced to "H" level	Start of counting	Counting possible
4	Counting not possible		Counting possible
5		(1) Start of counting (2) Retriggering	

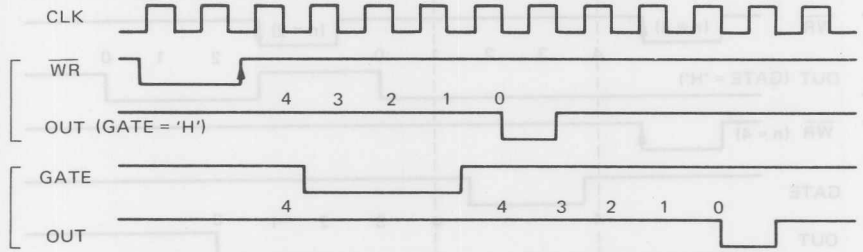




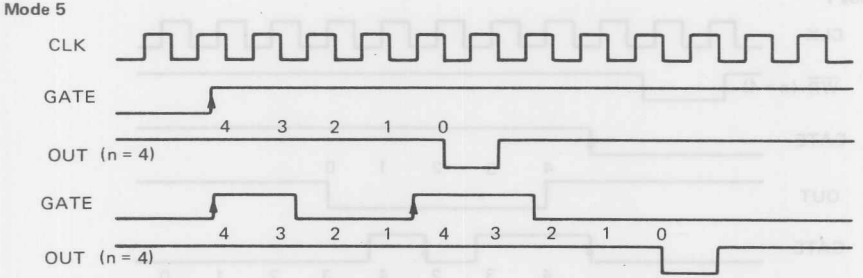
Mode 3



Mode 4



Mode 5



Note: "n" is the value set in the counter.
Figures in these diagrams refer to counter values.

Reading of Counter Values

All 82C53-5 counting is down-counting, the counting being in steps of 2 in mode 3. Counter values can be read during counting by (1) direct reading, and (2) counter latching ("read on the fly").

- **Direct reading**

  Counter values can be read by direct reading operations.

  Since the counter value read according to the timing of the $\overline{RD}$ and CLK signals is not guaranteed, it is necessary to stop the counting by a gate input signal, or to interrupt the clock input temporarily by an external circuit to ensure that the counter value is correctly read.

- **Counter latching**

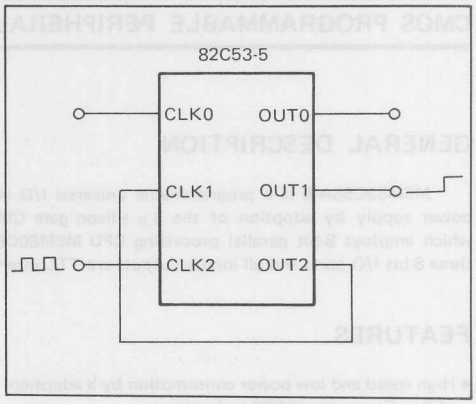
  In this method, the counter value is latched by writing a counter latch command, thereby enabling a stable value to be read without effecting the counting in any way at all. An example of a counter latching program is given below.

  Counter latching executed for counter #1 (Read/Load 2-byte setting)

```
MVI A  0 1 0 0 x x x x
                └──── Denotes counter latching
OUT n3 ──┬──── Write in control word address (n3)
         └──── The counter value at this point is latched
IN n1  ──┬──── Reading of the LSB of the counter value latched from counter #1.
         └──── n1: Counter #1 address
MOV B, A
IN n1  ──────── Reading of MSB from counter #1.
MOV C, A
```

Example of Practical Application

- **82C53-5 used as a 32-bit counter.**



Use counter #1 and counter #2

Counter #1: mode 0, upper order 16-bit counter value

Counter #2: mode 2, lower order 16-bit counter value

This setting enables counting up to a maximum of 2^{32}.

OKI semiconductor

MSM82C55A-5RS/GS

CMOS PROGRAMMABLE PERIPHERAL INTERFACE

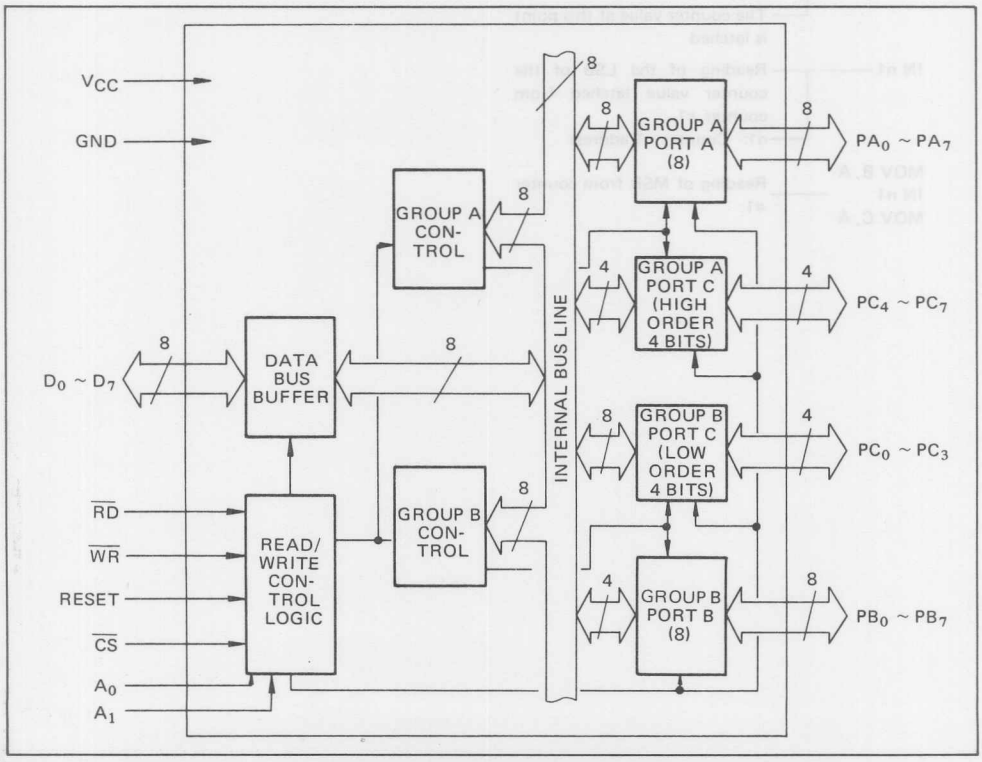
GENERAL DESCRIPTION

MSM82C55A-5 is a programmable universal I/O interface device which operates at a high speed and on a low power supply by adoption of the 3 μ silicon gate CMOS technology. It is the best fit as I/O port in a system which employs 8-bit parallel processing CPU MSM80C85A. Basically, this device has 24-bit I/O pins equivalent to three 8-bit I/O ports and all inputs/outputs are TTL interface compatible.

FEATURES

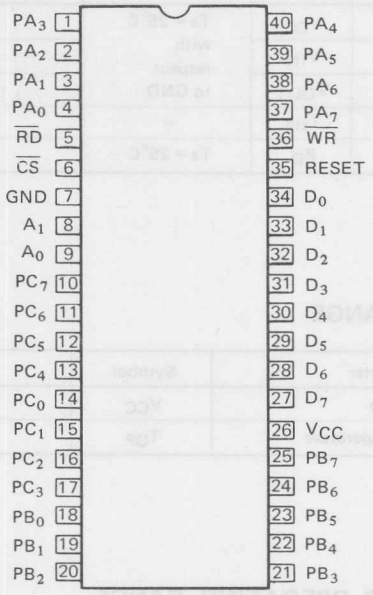
- High speed and low power consumption by a adoption of 3 μ silicon gate CMOS technology
- 3 V to 6 V single power supply
- Full static operation
- Programmable 24-bit I/O ports
- Bidirectional bus operation (Port A)
- Bit set/reset function (Port C)
- TTL compatible
- 40-pin DIP (MSM82C55A-5RS)
- 44-pin flat package (MSM82C55A-5GS)
- Compatible with 8255A-5

CIRCUIT CONFIGURATION

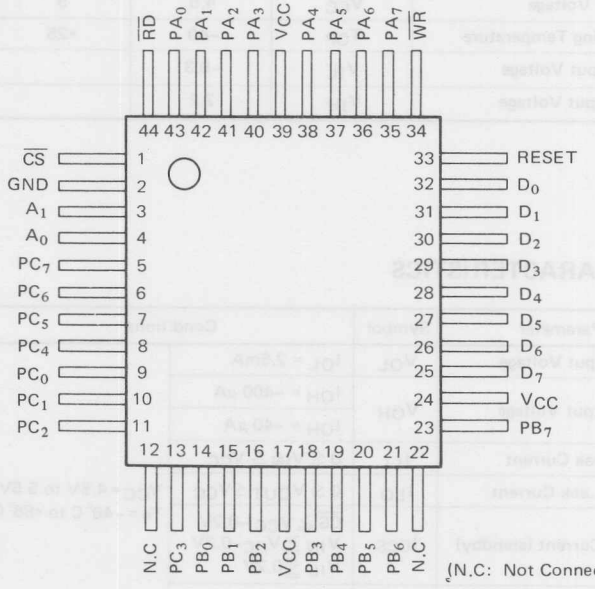


PIN CONFIGURATION

MSM82C55A-5RS (Top View)
40 Lead Plastic DIP



MSM82C55A-5GS (Top View)
44 Lead Plastic Flat Package



(N.C: Not Connected)

Note: A simplified form is used for model indication on the device in some cases.

Parameter	Symbol	Conditions	MSM82C55A-5RS	MSM82C55A-5GS	Unit
Supply Voltage	V_{CC}	Ta = 25°C with respect to GND	−0.5 to +7		V
Input Voltage	V_{IN}		−0.5 to V_{CC} +0.5		V
Output Voltage	V_{OUT}		−0.5 to V_{CC} +0.5		V
Storage Temperature	T_{stg}	—	−55 to +150		°C
Permissible Loss	P_D	Ta = 25°C	1.0	0.7	W

OPERATING RANGE

Parameter	Symbol	Limits	Unit
Supply Voltage	V_{CC}	3 to 6	V
Operating Temperature	T_{OP}	−40 to 85	°C

RECOMMENDED OPERATING RANGE

Parameter	Symbol	Min.	Typ.	Max.	Unit
Supply Voltage	V_{CC}	4.5	5	5.5	V
Operating Temperature	T_{OP}	−40	+25	+85	°C
"L" Input Voltage	V_{IL}	−0.3		+0.8	V
"H" Input Voltage	V_{IH}	2.2		V_{CC} +0.3	V

DC CHARACTERISTICS

Parameter	Symbol	Conditions		Min.	Typ.	Max.	Unit
"L" Output Voltage	V_{OL}	I_{OL} = 2.5mA	V_{CC} = 4.5V to 5.5V, Ta = −40° C to +85° C			0.45	V
"H" Output Voltage	V_{OH}	I_{OH} = −400 μA		2.4			V
		I_{OH} = −40 μA		4.2			V
Input Leak Current	I_{LI}	$0 \leq V_{IN} \leq V_{CC}$		−10		10	μA
Output Leak Current	I_{LO}	$0 \leq V_{OUT} \leq V_{CC}$		−10		10	μA
Supply Current (standby)	I_{CCS}	$\overline{CS} \geqq V_{CC}$ −0.2V $V_{IN} \geqq V_{CC}$−0.2V $V_{IL} \leqq 0.2V$			0.1	100	μA
Average Supply Current (active)	I_{CC}	I/O write cycle time: 1 μs				5	mA

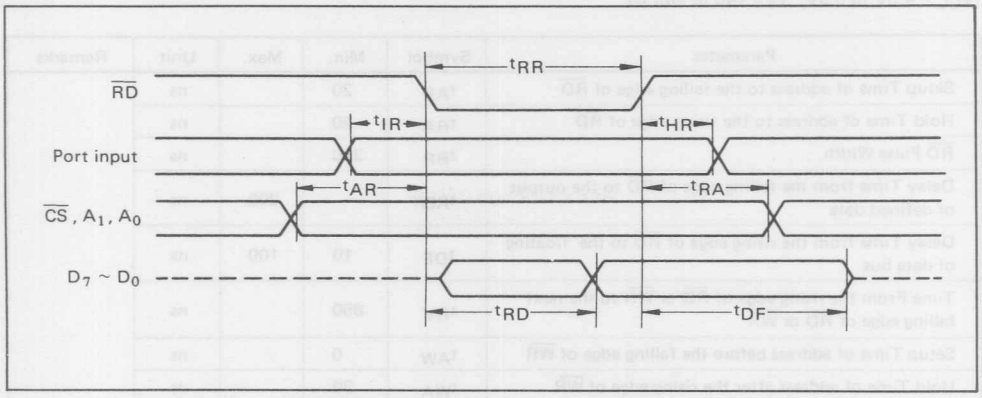
AC CHARACTERISTICS

(V_{CC} = 4.5V to 5.5V, Ta = –40 to +85°C)

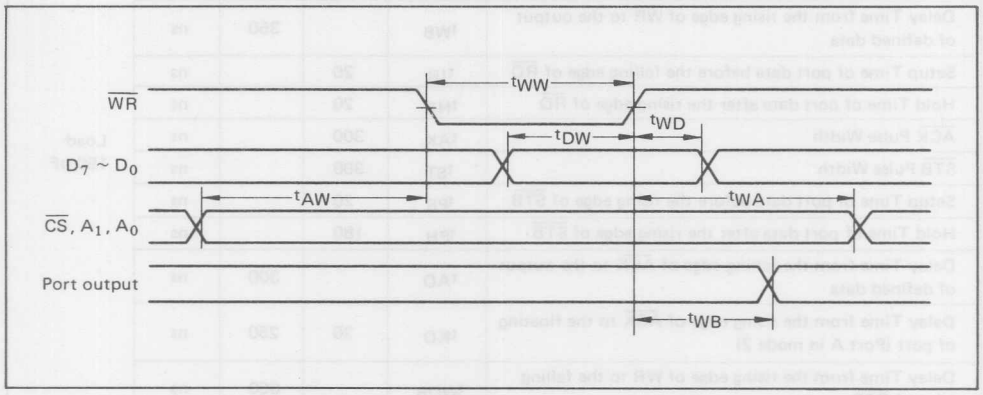
Parameter	Symbol	Min.	Max.	Unit	Remarks
Setup Time of address to the falling edge of $\overline{RD}$	t_{AR}	20		ns	
Hold Time of address to the rising edge of $\overline{RD}$	t_{RA}	20		ns	
$\overline{RD}$ Pulse Width	t_{RR}	300		ns	
Delay Time from the falling edge of $\overline{RD}$ to the output of defined data	t_{RD}		200	ns	
Delay Time from the rising edge of $\overline{RD}$ to the floating of data bus	t_{DF}	10	100	ns	
Time From the rising edge of $\overline{RD}$ or $\overline{WR}$ to the next falling edge of $\overline{RD}$ or $\overline{WR}$	t_{RV}	850		ns	
Setup Time of address before the falling edge of $\overline{WR}$	t_{AW}	0		ns	
Hold Time of address after the rising edge of $\overline{WR}$	t_{WA}	30		ns	
$\overline{WR}$ Pulse Width	t_{WW}	300		ns	
Setup Time of bus data before the rising edge of $\overline{WR}$	t_{DW}	100		ns	
Hold Time of bus data after the rising edge of $\overline{WR}$	t_{WD}	40		ns	
Delay Time from the rising edge of $\overline{WR}$ to the output of defined data	t_{WB}		350	ns	
Setup Time of port data before the falling edge of $\overline{RD}$	t_{IR}	20		ns	
Hold Time of port data after the rising edge of $\overline{RD}$	t_{HR}	20		ns	
$\overline{ACK}$ Pulse Width	t_{AK}	300		ns	Load 150 pF
$\overline{STB}$ Pulse Width	t_{ST}	300		ns	
Setup Time of port data before the rising edge of $\overline{STB}$	t_{PS}	20		ns	
Hold Time of port data after the rising edge of $\overline{STB}$	t_{PH}	180		ns	
Delay Time from the falling edge of $\overline{ACK}$ to the output of defined data	t_{AD}		300	ns	
Delay Time from the rising edge of $\overline{ACK}$ to the floating of port (Port A in mode 2)	t_{KD}	20	250	ns	
Delay Time from the rising edge of WR to the falling edge of $\overline{OBF}$	t_{WOB}		650	ns	
Delay Time from the falling edge of $\overline{ACK}$ to the rising edge of $\overline{OBF}$	t_{AOB}		350	ns	
Delay Time from the falling edge of $\overline{STB}$ to the rising edge of IBF	t_{SIB}		300	ns	
Delay Time from the rising edge of $\overline{RD}$ to the falling edge of IBF	t_{RIB}		300	ns	
Delay Time from the falling edge of $\overline{RD}$ to the falling edge of INTR	t_{RIT}		400	ns	
Delay Time from the rising edge of $\overline{STB}$ to the rising edge of INTR	t_{SIT}		300	ns	
Delay Time from the rising edge of $\overline{ACK}$ to the rising edge of INTR	t_{AIT}		350	ns	
Delay Time from the falling edge of $\overline{WR}$ to the falling edge of INTR	t_{WIT}		850	ns	

Note: Timing is measured at V_L = 0.8 V and V_H = 2.2 V for both inputs and outputs.

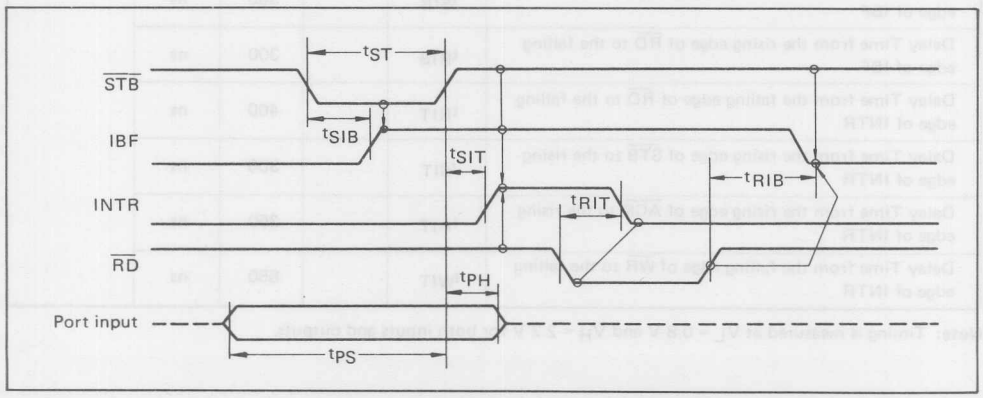
Basic Input Operation (Mode 0)



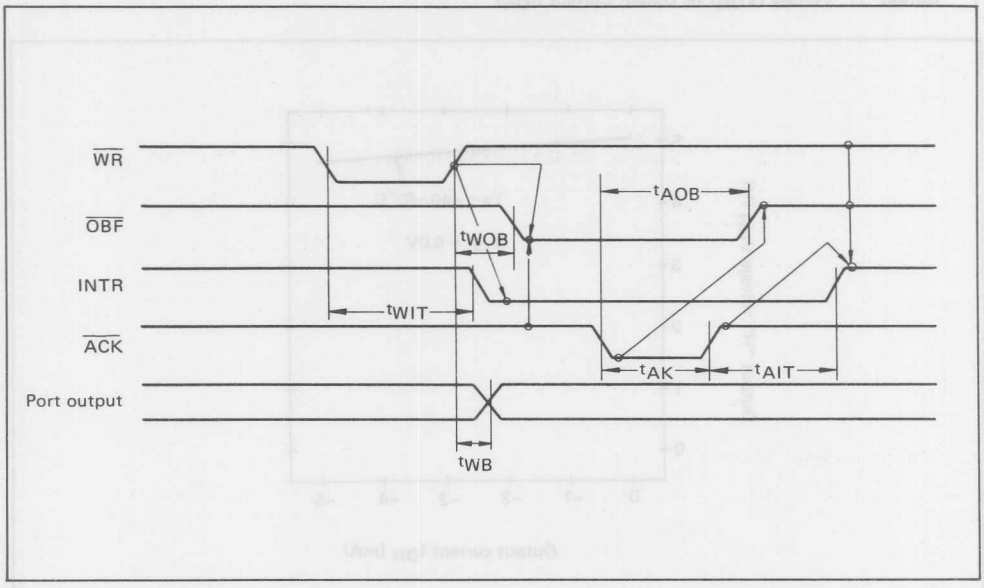
Basic Output Operation (Mode 0)



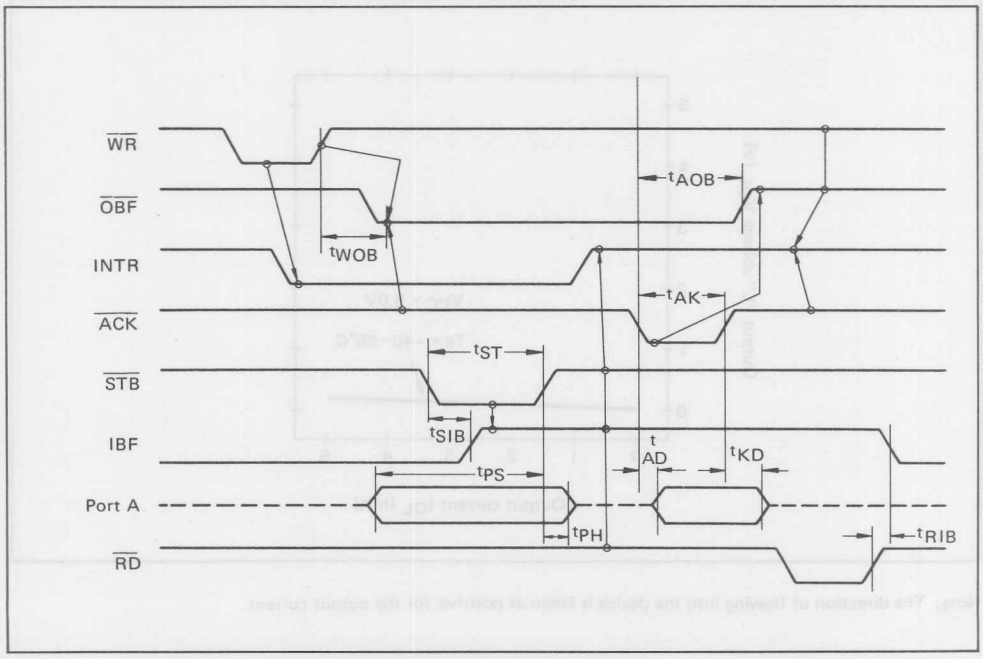
Strobe Input Operation (Mode 1)

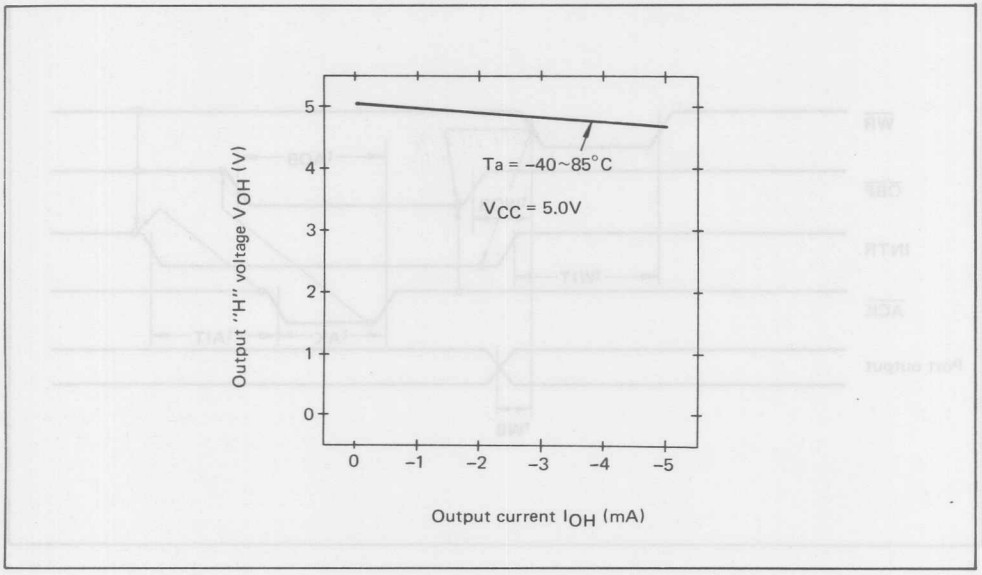


Strobe Output Operation (Mode 1)

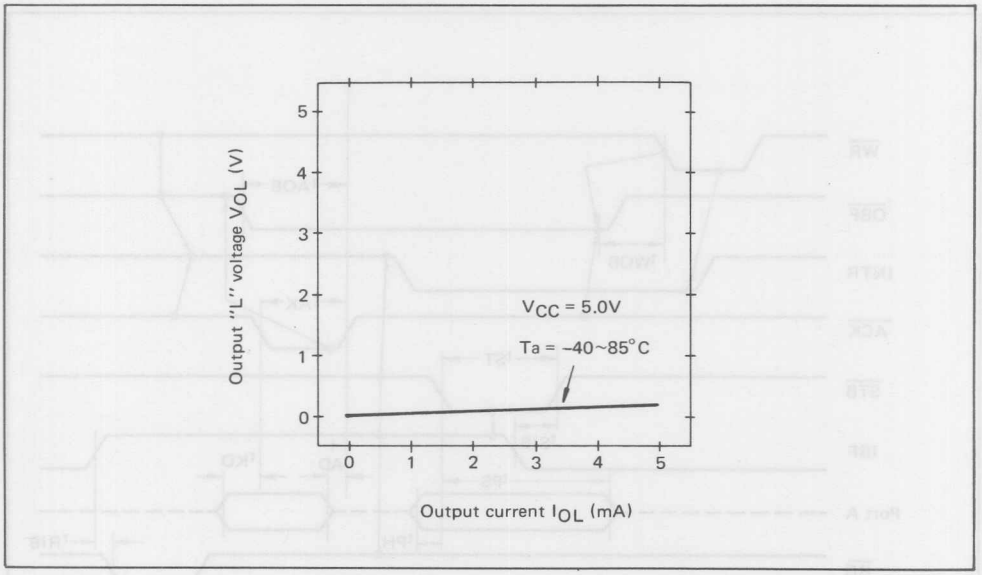


Bidirectional Bus Operation (Mode 2)





2 Output "L" Voltage (V_{OL}) vs. Output Current (I_{OL})



Note: The direction of flowing into the device is taken as positive for the output current.

FUNCTIONAL DESCRIPTION OF PIN

Pin No.	Item	Input/Output	Function
D7 ~ D0	Bidirectional data bus	Input and output	These are three-state 8-bit bidirectional buses used to write and read data upon receipt of the $\overline{WR}$ and $\overline{RD}$ signals from CPU and also used when control words and bit set/reset data are transferred from CPU to MSM82C55A-5.
RESET	Reset input	Input	This signal is used to reset the control register and all internal registers when it is in high level. At this time, ports are all made into the input mode (high impedance status).
$\overline{CS}$	Chip select input	Input	When the $\overline{CS}$ is in low level, data transmission is enabled with CPU. When it is in high level, the data bus is made into the high impedance status where no write nor read operation is performed. Internal registers hold their previous status, however.
$\overline{RD}$	Read input	Input	When RD is in low level, data is transferred from MSM82C55A-5 to CPU.
$\overline{WR}$	Write input	Input	When WR is in low level, data or control words are transferred from CPU to MSM82C55A-5.
A0, A1	Port select input (address)	Input	By combination of A0 and A1, either one is selected from among port A, port B, port C, and control register. These pins are usually connected to low order 2 bits of the address bus.
PA7 ~ PA0	Port A	Input and output	These are universal 8-bit I/O ports. The direction of inputs/outputs can be determined by writing a control word. Especially, port A can be used as a bidirectional port when it is set to mode 2.
PB7 ~ PB0	Port B	Input and output	These are universal 8-bit I/O ports. The direction of inputs/outputs can be determined by writing a control word.
PC7 ~ PC0	Port C	Input and output	These are universal 8-bit I/O ports. The direction of inputs/outputs can be determined by writing a control word as 2 ports with 4 bits each. When port A or port B is used in mode 1 or mode 2 (port A only), they become control pins. Especially when port C is used as an output port, each bit can be set/reset independently.
V_{CC}			+5 V power supply.
GND			GND

BASIC FUNCTIONAL DESCRIPTION

Group A and Group B

When setting a mode to a port having 24 bits, set it by dividing it into two groups of 12 bits each.

Group A: Port A (8 bits) and high order 4 bits of port C (PC7 ~ PC4)

Group B: Port B (8 bits) and low order 4 bits of port C (PC3 ~ PC0)

Mode 0, 1, 2

There are 3 types of modes to be set by group as follows:

Mode 0: Basic input operation/output operation (Available for both groups A and B)

Mode 1: Strobe input operation/output operation (Available for both groups A and B)

Mode 2: Bidirectional bus operation (Available for group A only)

When used in mode 1 or mode 2, however, port C has bits to be defined as ports for control signal for operation ports (port A for group A and port B for group B) of their respective groups.

Port A, B, C

The internal structure of 3 ports is as follows:

Port A: One 8-bit data output latch/buffer and one 8-bit data input latch

Port B: One 8-bit data input/output latch/buffer and one 8-bit data input buffer

Port C: One 8-bit data output latch/buffer and one 8-bit data input buffer (no latch for input)

Single bit set/reset function for port C

When port C is defined as output port, it is possible to set (to turn to high level) or reset (to turn to low level) any one of 8 bits individually without affecting other bits.

OPERATIONAL DESCRIPTION

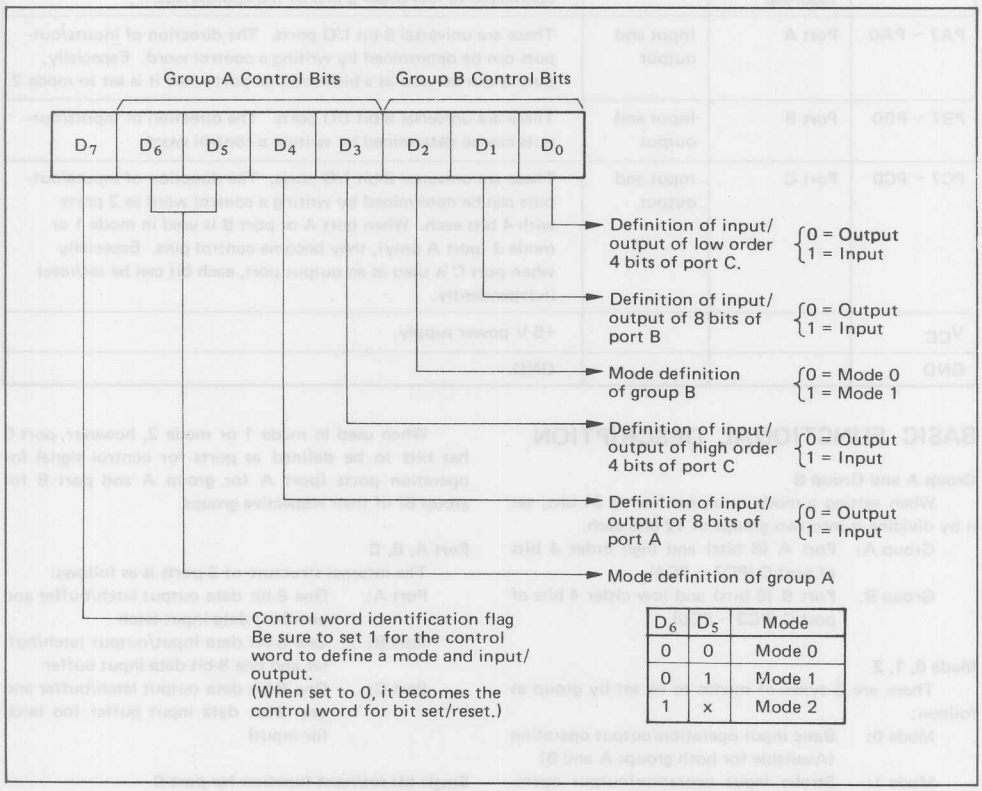
Control Logic

Operations by addresses and control signals, e.g., read and write, etc. are as shown in the table below:

Operation	A1	A0	$\overline{CS}$	$\overline{WR}$	$\overline{RD}$	Operation
Input	0	0	0	1	0	Port A → Data Bus
	0	1	0	1	0	Port B → Data Bus
	1	0	0	1	0	Port C → Data Bus
Output	0	0	0	0	1	Data Bus → Port A
	0	1	0	0	1	Data Bus → Port B
	1	0	0	0	1	Data Bus → Port C
Control	1	1	0	0	1	Data Bus → Control Register
Others	1	1	0	1	0	Illegal Condition
	x	x	1	x	x	Data bus is in the high impedance status.

Setting of Control Word

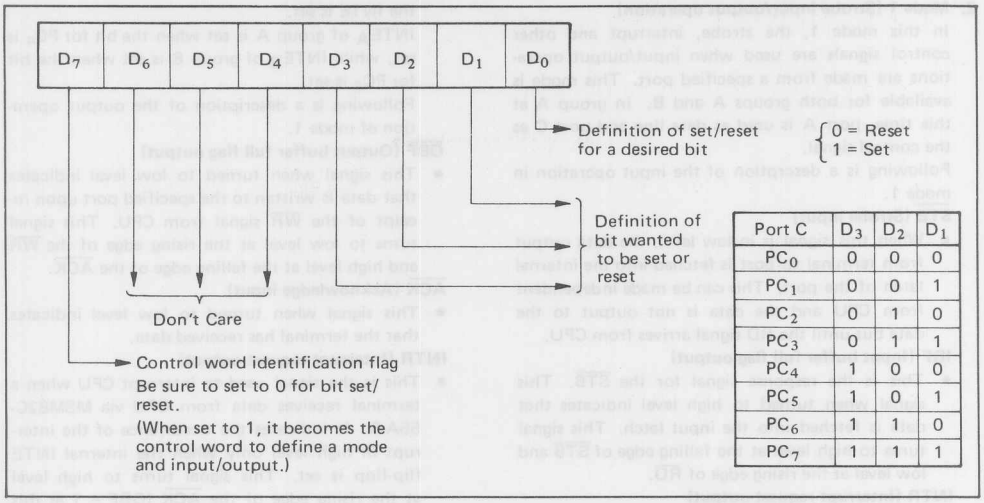
The control register is composed of 7-bit latch circuit and 1-bit flag as shown below.



D_6	D_5	Mode
0	0	Mode 0
0	1	Mode 1
1	x	Mode 2

Bit Set/Reset Function

When port C is defined as output port, it is possible to set (set output to 1) or reset (set output to 0) any one of 8 bits without affecting other bits as shown next page.



Port C	D_3	D_2	D_1
PC_0	0	0	0
PC_1	0	0	1
PC_2	0	1	0
PC_3	0	1	1
PC_4	1	0	0
PC_5	1	0	1
PC_6	1	1	0
PC_7	1	1	1

Interrupt Control Function

When MSM82C55A-5 is used in mode 1 or mode 2, the interrupt signal for CPU is provided. The interrupt request signal is output from port C. When the internal flip-flop INTE is set beforehand at this time, the desired interrupt request signal is output. When it is reset beforehand, however, the interrupt request signal is not output. The set/reset of the internal flip-flop is made by the bit set/reset operation for port C virtually.

Bit set → INTE is set → Interrupt allowed

Bit reset → INTE is reset → Interrupt inhibited

Operational Description by Mode

1. Mode 0 (Basic input/output operation)

Mode 0 makes MSM82C55A-5 operate as a basic input port or output port. As no control signal such as interrupt request, etc. is required in this mode. All of 24 bits can be used as two-8-bit ports and two 4-bit ports. Sixteen combinations are then possible for inputs/outputs. The inputs are not latched, but the outputs are.

Type	Control Word								Group A		Group B	
	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Port A	High Order 4 Bits of Port C	Port B	Low Order 4 Bits of Port C
1	1	0	0	0	0	0	0	0	Output	Output	Output	Output
2	1	0	0	0	0	0	0	1	Output	Output	Output	Input
3	1	0	0	0	0	0	1	0	Output	Output	Input	Output
4	1	0	0	0	0	0	1	1	Output	Output	Input	Input
5	1	0	0	0	1	0	0	0	Output	Input	Output	Output
6	1	0	0	0	1	0	0	1	Output	Input	Output	Input
7	1	0	0	0	1	0	1	0	Output	Input	Input	Output
8	1	0	0	0	1	0	1	1	Output	Input	Input	Input
9	1	0	0	1	0	0	0	0	Input	Output	Output	Output
10	1	0	0	1	0	0	0	1	Input	Output	Output	Input
11	1	0	0	1	0	0	1	0	Input	Output	Input	Output
12	1	0	0	1	0	0	1	1	Input	Output	Input	Input
13	1	0	0	1	1	0	0	0	Input	Input	Output	Output
14	1	0	0	1	1	0	0	1	Input	Input	Output	Input
15	1	0	0	1	1	0	1	0	Input	Input	Input	Output
16	1	0	0	1	1	0	1	1	Input	Input	Input	Input

Note: When used in mode 0 for both groups A and B

2. Mode 1 (Strobe input/output operation)

In this mode 1, the strobe, interrupt and other control signals are used when input/output operations are made from a specified port. This mode is available for both groups A and B. In group A at this time, port A is used as data line and port C as the control signal.

Following is a descrption of the input operation in mode 1.

$\overline{STB}$ (Strobe input)

- When this signal is in low level, the data output from terminal to port is fetched into the internal latch of the port. This can be made independent from CPU and the data is not output to the data bus until the RD signal arrives from CPU.

IBF (Input buffer full flag output)

- This is the response signal for the $\overline{STB}$. This signal when turned to high level indicates that data is fetched into the input latch. This signal turns to high level at the falling edge of $\overline{STB}$ and low level at the rising edge of $\overline{RD}$.

INTR (Interrupt request output)

- This is the interrupt request signal for CPU of the data fetched into the input latch. It is indicated by high level only when the internal INTE flip-flop is set. This signal turns to high level at the rising edge of the $\overline{STB}$ (IBF = 1 at this time) and low level at the falling edge of the $\overline{RD}$ when the INTE is set.

$INTE_A$ of group A is set when the bit for PC_4 is set, while $INTE_B$ of group B is set when the bit for PC_2 is set.

Following is a description of the output operation of mode 1.

$\overline{OBF}$ (Output buffer full flag output)

- This signal when turned to low level indicates that data is written to the specified port upon receipt of the $\overline{WR}$ signal from CPU. This signal turns to low level at the rising edge of the $\overline{WR}$ and high level at the falling edge of the $\overline{ACK}$.

$\overline{ACK}$ (Acknowledge input)

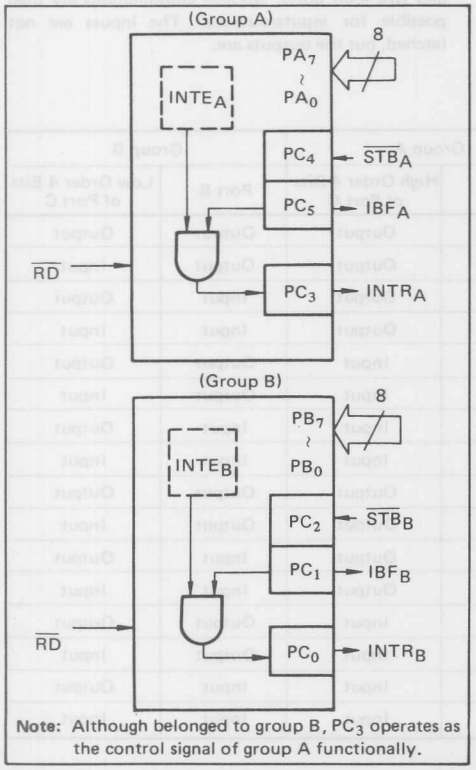
- This signal when turned to low level indicates that the terminal has received data.

INTR (Interrupt request output)

- This is the signal used to interrupt CPU when a terminal receives data from CPU via MSM82C-55A-5. It indicates the occurrence of the interrupt in high level only when the internal INTE flip-flop is set. This signal turns to high level at the rising edge of the $\overline{ACK}$ (OBF = 1 at this time) and low level at the falling edge of $\overline{WR}$ when the $INTE_B$ is set.

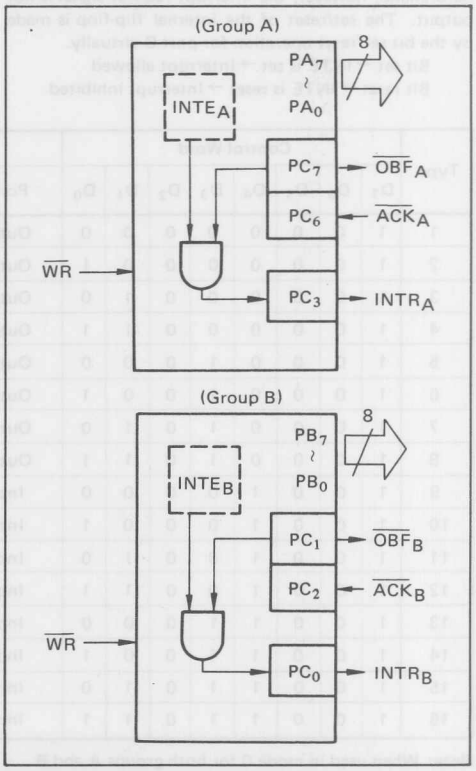
$INTE_A$ of group A is set when the bit for PC_6 is set, while $INTE_B$ of group B is set when the bit for PC_2 is set.

Mode 1 Input



Note: Although belonged to group B, PC_3 operates as the control signal of group A functionally.

Mode 1 output



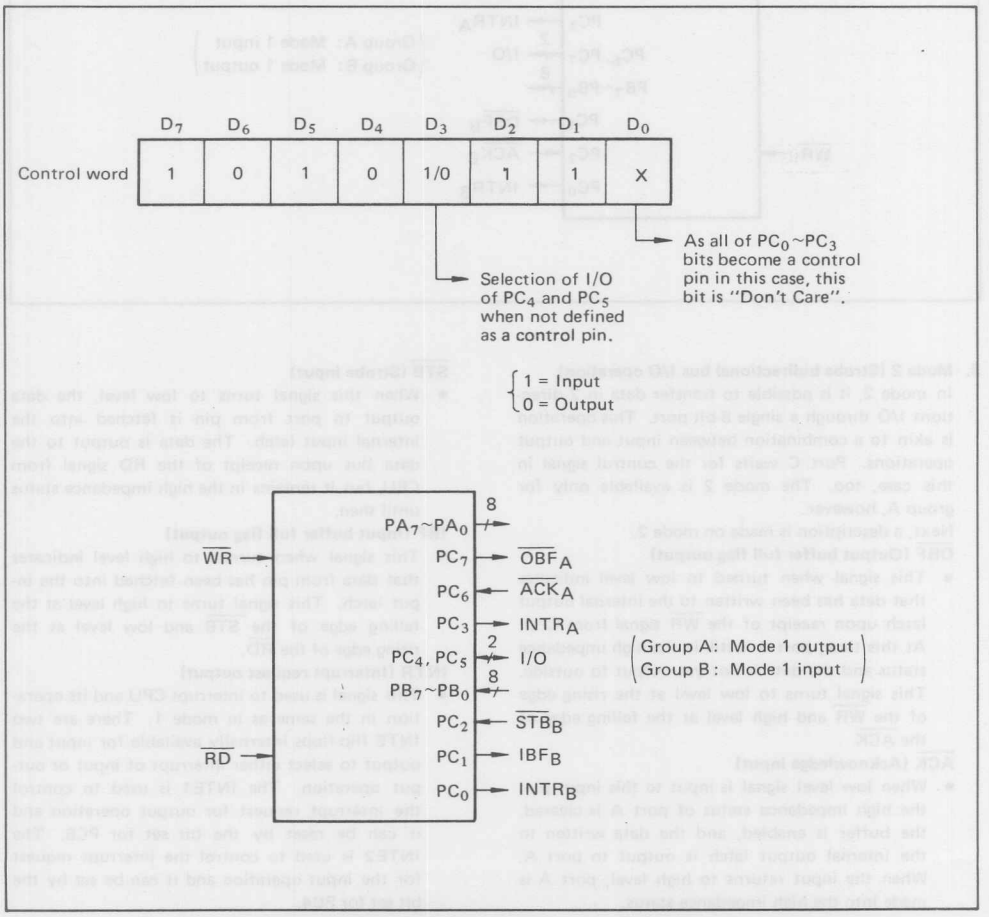
Port C Function Allocation in Mode 1

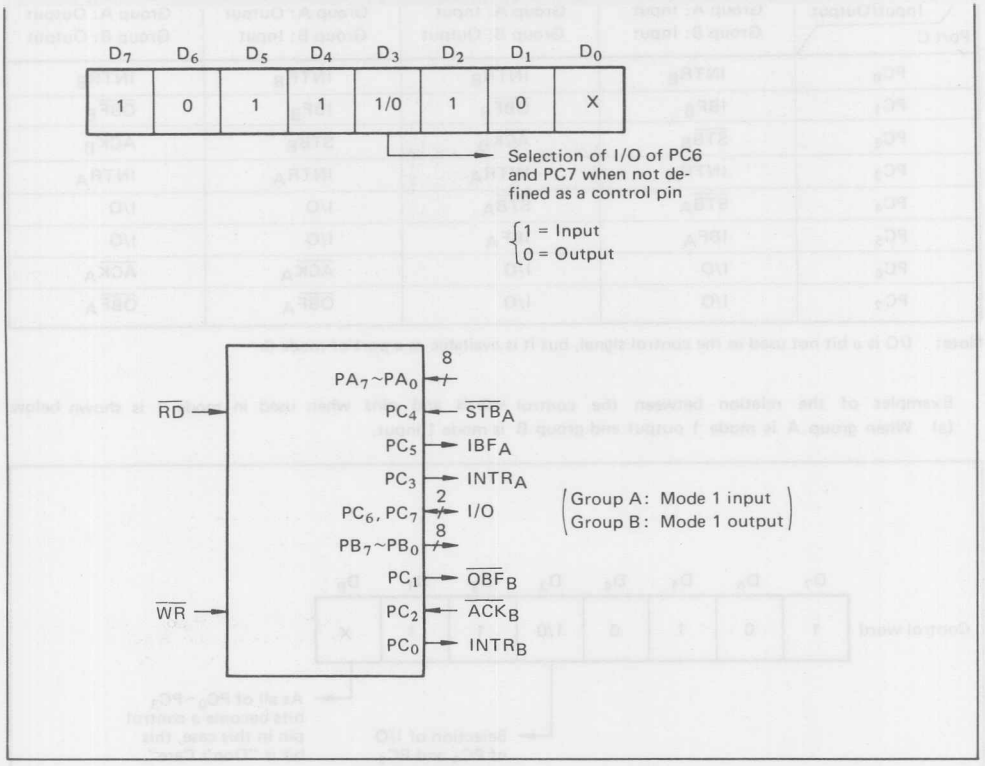
Combination of Input/Output / Port C	Group A: Input Group B: Input	Group A: Input Group B: Output	Group A: Output Group B: Input	Group A: Output Group B: Output
PC_0	$INTR_B$	$INTR_B$	$INTR_B$	$INTR_B$
PC_1	IBF_B	$\overline{OBF}_B$	IBF_B	$\overline{OBF}_B$
PC_2	$\overline{STB}_B$	$\overline{ACK}_B$	$\overline{STB}_B$	$\overline{ACK}_B$
PC_3	$INTR_A$	$INTR_A$	$INTR_A$	$INTR_A$
PC_4	$\overline{STB}_A$	$\overline{STB}_A$	I/O	I/O
PC_5	IBF_A	IBF_A	I/O	I/O
PC_6	I/O	I/O	$\overline{ACK}_A$	$\overline{ACK}_A$
PC_7	I/O	I/O	$\overline{OBF}_A$	$\overline{OBF}_A$

Note: I/O is a bit not used as the control signal, but it is available as a port of mode 0.

Examples of the relation between the control words and pins when used in mode 1 is shown below:

(a) When group A is mode 1 output and group B is mode 1 input.





3. Mode 2 (Strobe bidirectional bus I/O operation)

In mode 2, it is possible to transfer data in 2 directions I/O through a single 8-bit port. This operation is akin to a combination between input and output operations. Port C waits for the control signal in this case, too. The mode 2 is available only for group A, however.

Next, a description is made on mode 2.

$\overline{OBF}$ (Output buffer full flag output)

- This signal when turned to low level indicates that data has been written to the internal output latch upon receipt of the $\overline{WR}$ signal from CPU. At this time, port A is still in the high impedance status and the data is not yet output to outside. This signal turns to low level at the rising edge of the $\overline{WR}$ and high level at the falling edge of the $\overline{ACK}$

$\overline{ACK}$ (Acknowledge input)

- When low level signal is input to this input pin, the high impedance status of port A is cleared, the buffer is enabled, and the data written to the internal output latch is output to port A. When the input returns to high level, port A is made into the high impedance status.

$\overline{STB}$ (Strobe input)

- When this signal turns to low level, the data output to port from pin is fetched into the internal input latch. The data is output to the data bus upon receipt of the RD signal from CPU, but it remains in the high impedance status until then.

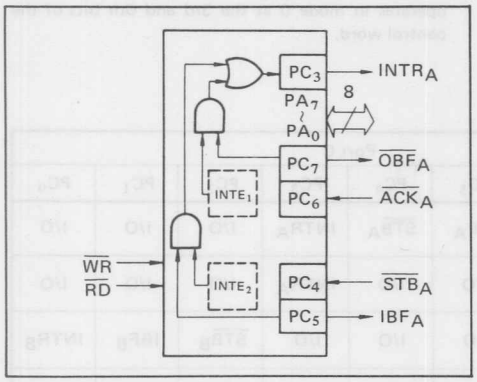
IBF (Input buffer full flag output)

- This signal when turned to high level indicates that data from pin has been fetched into the input latch. This signal turns to high level at the falling edge of the $\overline{STB}$ and low level at the rising edge of the $\overline{RD}$.

INTR (Interrupt request output)

- This signal is used to interrupt CPU and its operation in the same as in mode 1. There are two INTE flip-flops internally available for input and output to select either interrupt of input or output operation. The INTE1 is used to control the interrupt request for output operation and it can be reset by the bit set for PC6. The INTE2 is used to control the interrupt request for the input operation and it can be set by the bit set for PC4.

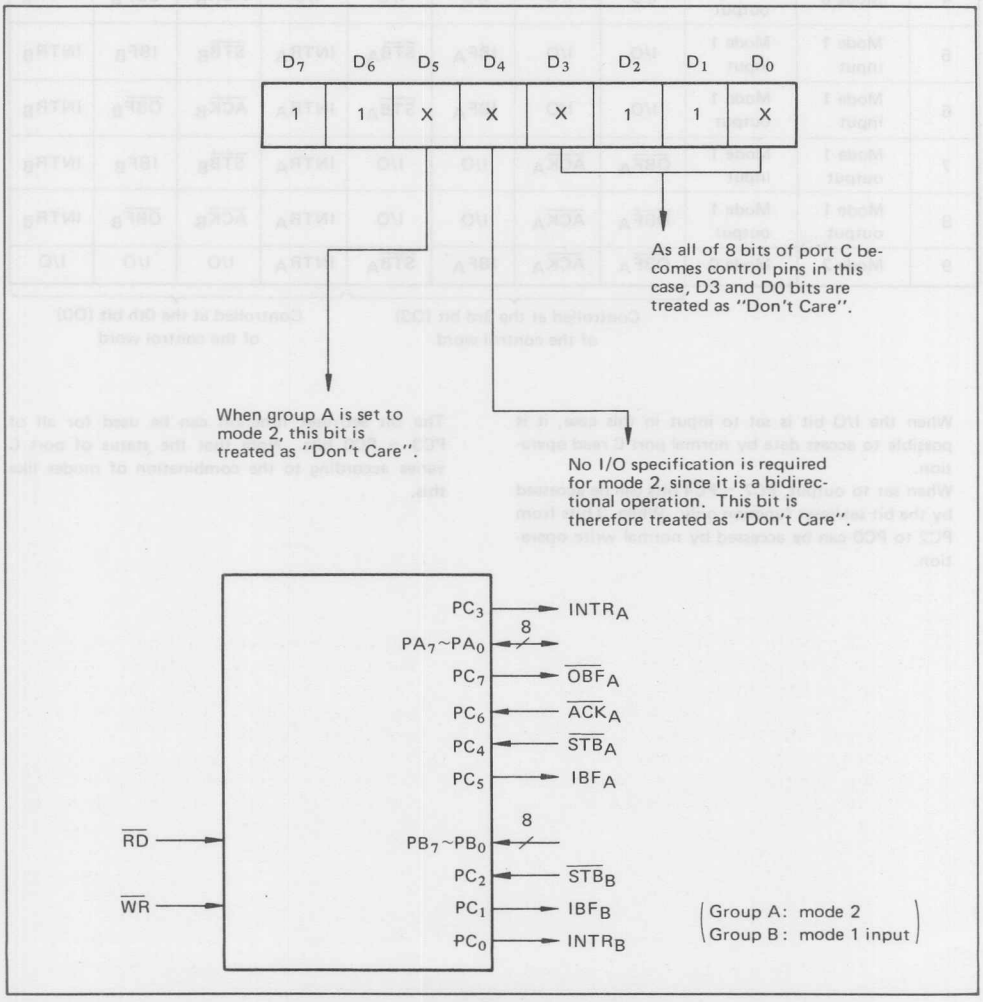
Mode 2 I/O Operation



Port C Function Allocation in Mode 2

Port C	Function
PC_0	Confirmed to the group B mode
PC_1	
PC_2	
PC_3	$INTR_A$
PC_4	$\overline{STB}_A$
PC_5	IBF_A
PC_6	$\overline{ACK}_A$
PC_7	$\overline{OBF}_A$

Following is an example of the relation between the control word and pin when used in mode 2. When input in mode 2 for group A and in mode 1 for group B.



4. When Group A is Different in Mode from Group B

Group A and group B can be used by setting them in different modes each other at the same time. When either group is set to mode1 or mode 2, it is possible to set the one not defined as control pin in port C to both input and output as a port which operates in mode 0 at the 3rd and 0th bits of the control word.

(mode conbinations that define no control bit at port C)

	Group A	Group B	Port C							
			PC_7	PC_6	PC_5	PC_4	PC_3	PC_2	PC_1	PC_0
1	Mode 1 input	Mode 0	I/O	I/O	IBF_A	$\overline{STB}_A$	$INTR_A$	I/O	I/O	I/O
2	Mode 0 output	Mode 0	$\overline{OBF}_A$	$\overline{ACK}_A$	I/O	I/O	$INTR_A$	I/O	I/O	I/O
3	Mode 0	Mode 1 input	I/O	I/O	I/O	I/O	I/O	$\overline{STB}_B$	IBF_B	$INTR_B$
4	Mode 0	Mode 1 output	I/O	I/O	I/O	I/O	I/O	$\overline{ACK}_B$	$\overline{OBF}_B$	$INTR_B$
5	Mode 1 input	Mode 1 input	I/O	I/O	IBF_A	$\overline{STB}_A$	$INTR_A$	$\overline{STB}_B$	IBF_B	$INTR_B$
6	Mode 1 input	Mode 1 output	I/O	I/O	IBF_A	$\overline{STB}_A$	$INTR_A$	$\overline{ACK}_B$	$\overline{OBF}_B$	$INTR_B$
7	Mode 1 output	Mode 1 input	$\overline{OBF}_A$	$\overline{ACK}_A$	I/O	I/O	$INTR_A$	$\overline{STB}_B$	IBF_B	$INTR_B$
8	Mode 1 output	Mode 1 output	$\overline{OBF}_A$	$\overline{ACK}_A$	I/O	I/O	$INTR_A$	$\overline{ACK}_B$	$\overline{OBF}_B$	$INTR_B$
9	Mode 2	Mode 0	$\overline{OBF}_A$	$\overline{ACK}_A$	IBF_A	$\overline{STB}_A$	$INTR_A$	I/O	I/O	I/O

Controlled at the 3rd bit (D3) of the control word

Controlled at the 0th bit (D0) of the control word

When the I/O bit is set to input in this case, it is possible to access data by normal port C read operation.

When set to output, PC7 ~ PC4 bits can be accessed by the bit set/reset function only. While, 3 bits from PC2 to PC0 can be accessed by normal write operation.

The bit set/reset function can be used for all of PC3 ~ PC0 bits. Note that the status of port C varies according to the combination of modes like this.

5. Port C Status Read

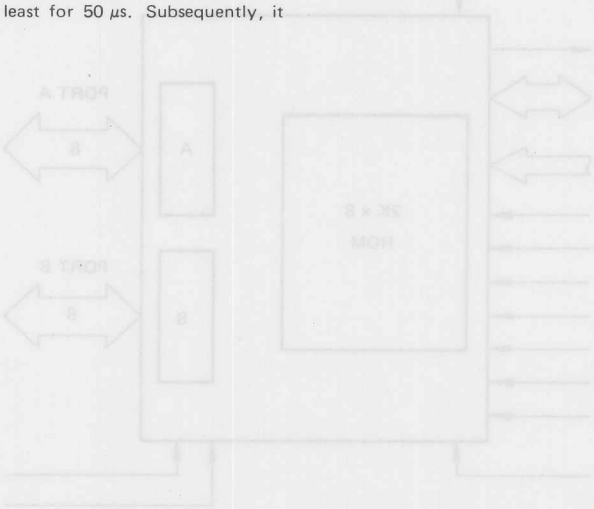
When port C is used for the control signal, that is, in either mode 1 or mode 2, each control signal and bus status signal can be read out by reading the content of port C.

The status read out is as follows:

	Group A	Group B	Status read on the data bus							
			D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	Mode 1 input	Mode 0	I/O	I/O	IBF_A	$INTE_A$	$INTR_A$	I/O	I/O	I/O
2	Mode 1 output	Mode 0	$\overline{OBF}_A$	$INTE_A$	I/O	I/O	$INTR_A$	I/O	I/O	I/O
3	Mode 0	Mode 1 input	I/O	I/O	I/O	I/O	I/O	$INTE_B$	IBF_B	$INTR_B$
4	Mode 0	Mode 1 output	I/O	I/O	I/O	I/O	I/O	$INTE_B$	$\overline{OBF}_B$	$INTR_B$
5	Mode 1 input	Mode 1 input	I/O	I/O	IBF_A	$INTE_A$	$INTR_A$	$INTE_B$	IBF_B	$INTR_B$
6	Mode 1 input	Mode 1 output	I/O	I/O	IBF_A	$INTE_A$	$INTR_A$	$INTE_B$	$\overline{OBF}_B$	$INTR_B$
7	Mode 1 output	Mode 1 input	$\overline{OBF}_A$	$INTE_A$	I/O	I/O	$INTR_A$	$INTE_B$	IBF_B	$INTR_B$
8	Mode 1 output	Mode 1 output	$\overline{OBF}_A$	$INTE_A$	I/O	I/O	$INTR_A$	$INTE_B$	$\overline{OBF}_B$	$INTR_B$
9	Mode 2	Mode 0	$\overline{OBF}_A$	$INTE_1$	IBF_A	$INTE_2$	$INTR_A$	I/O	I/O	I/O
10	Mode 2	Mode 1 input	$\overline{OBF}_A$	$INTE_1$	IBF_A	$INTE_2$	$INTR_A$	$INTE_B$	IBF_B	$INTR_B$
11	Mode 2	Mode 1 output	$\overline{OBF}_A$	$INTE_1$	IBF_A	$INTE_2$	$INTR_A$	$INTE_B$	$\overline{OBF}_B$	$INTR_B$

6. Reset of MSM82C55A-5

Be sure to keep the RESET signal at power ON in the high level at least for 50 μs. Subsequently, it becomes the input mode at a high level pulse above 500 ns.



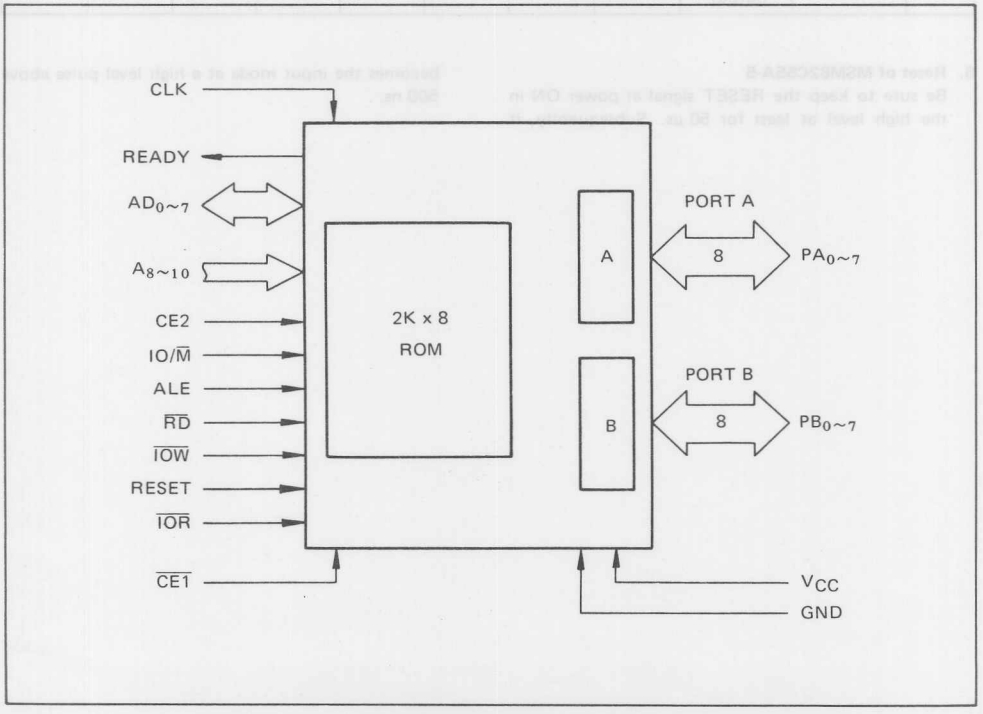
GENERAL DESCRIPTION

MSM83C55 is a combination of ROM and I/O devices used in a microcomputer system. Owing to the adoption of the CMOS silicon gate technology, it operates on a low power supply as small as 100 μA (max.) in the standby current in the chip non-select status. As the ROM is composed of 2048 words x 8 bits and its access time (max.) is 400 ns, it can be applied without using the wait state in the 80C85A system, too. The I/O circuit is composed of 2 universal I/O ports. Each of these I/O ports has 8 port lines and each of these port lines can be programmed as input or output line independently.

FEATURES

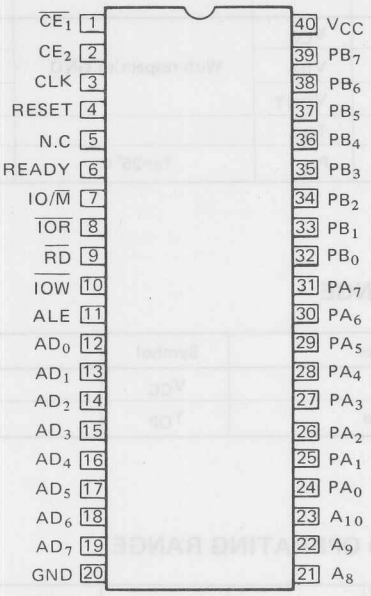
- High speed and low power consumption owing to adoption of silicon gate CMOS
- Composed of 2048 words x 8 bits
- 3 ~ 6 V single power supply
- Address latch circuit incorporated
- Provided with 2 universal 8-bit I/O ports
- Indivisual I/O port line programmable as input or output
- Time division address/data bus
- 40-pin DIP (MSM83C55-xxRS)
- 44-pin flat package (MSM83C55-xxGS)

CIRCUIT CONFIGURATION

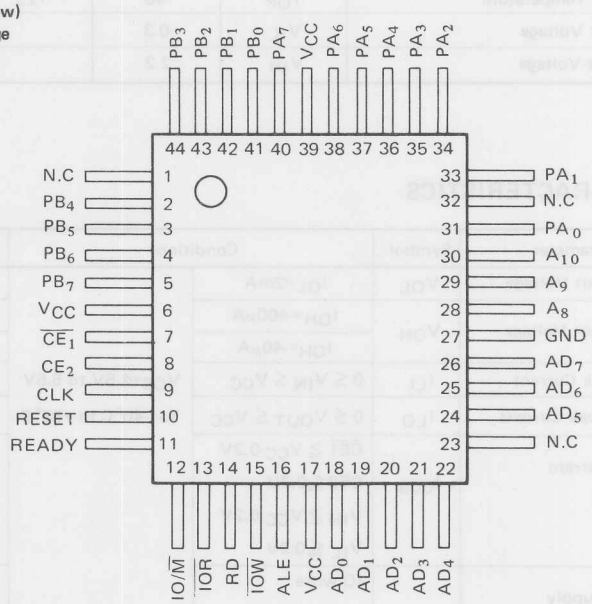


PIN CONFIGURATION

MSM83C55-xxRS (Top View)
40 Lead Plastic DIP



MSM83C55-xxGS (Top View)
44 Lead Plastic Flat Package



ABSOLUTE MAXIMUM RATINGS

Parameter	Symbol	Conditions	Limits		Unit
			MSM83C55RS	MSM83C55GS	
Supply Voltage	V_{CC}	With respect to GND	-0.5 to +7		V
Input Voltage	V_{IN}		-0.5 to V_{CC} +0.5		V
Output Voltage	V_{OUT}		-0.5 to V_{CC} +0.5		V
Storage Temperature	Tstg		-55 to +15D		°C
Permissible	P_D	Ta=25°C	1.0	0.7	W

OPERATING RANGE

Parameter	Symbol	Limits	Unit
Supply Voltage	V_{CC}	3 to 6	V
Operating Temperature	T_{OP}	-40 to +85	°C

RECOMMENDED OPERATING RANGE

Parameter	Symbol	Min.	Typ.	Max.	Unit
Supply Voltage	V_{CC}	4.5	5	5.5	V
Operating Temperature	T_{OP}	-40	+25	+85	°C
"L" Input Voltage	V_{IL}	-0.3		+0.8	V
"H" Input Voltage	V_{IH}	2.2		V_{CC} +0.3	V

DC CHARACTERISTICS

Parameter	Symbol	Conditions		Min.	Typ.	Max.	Unit
"L" Output Voltage	V_{OL}	I_{OL}=2mA	V_{CC}=4.5V to 5.5V Ta=-40°C to +85°C			0.45	V
"H" Output Voltage	V_{OH}	I_{OH}=-400μA		2.4			V
		I_{OH}=-40μA		4.2			V
Input Leak Current	I_{LI}	$0 \leq V_{IN} \leq V_{CC}$		-10		10	μA
Output Leak Current	I_{LO}	$0 \leq V_{OUT} \leq V_{CC}$		-10		10	μA
Supply Current (standby)	I_{CCS}	$\overline{CE1} \geqq V_{CC}$-0.2V CE2≦0.2V $V_{IH} \geqq V_{CC}$-0.2V $V_{IL} \leqq$0.2V			0.1	100	μA
Average Supply Current (active)	I_{CC}	IO write cycle time: 1 μs				5	mA

AC CHARACTERISTICS

(V_{CC}=4.5V to 5.5V, Ta=-40°C to +85°C)

Parameter	Symbol	Min.	Max.	Unit
Clock Cycle Time	t_{CYC}	320		ns
Clock Pulse Width	T_1	80		ns
Clock Pulse Width	T_2	120		ns
Clock Rise and Fall Time	t_f, t_r		30	ns
Address to Latch Setup Time	t_{AL}	50		ns
Address Hold Time after Latch	t_{LA}	30		ns
Latch to READ/WRITE Control	t_{LC}	100		ns
Valid Data Out Delay from READ Control	t_{RD}		170	ns
Address Stable to Data Out Valid	t_{AD}		400	ns
Latch Enable Width	t_{LL}	100		ns
Data Bus Float after READ	t_{RDF}	0	100	ns
READ/WRITE Control to Latch Enable	t_{CL}	20		ns
READ/WRITE Control Width	t_{CC}	250		ns
Data In to WRITE Setup Time	t_{DW}	150		ns
Data In Hold Time after WRITE	t_{WD}	10		ns
WRITE to Port Output	t_{WP}		400	ns
Port Input Setup Time	t_{PR}	50		ns
Port Input Hold Time	t_{RP}	50		ns
READY Hold Time	t_{PYH}	0	160	ns
Address to READY	t_{ARY}		160	ns
Recovery Time between Controls	t_{RV}	300		ns
Data Out Delay from READ Control	t_{RDE}	10		ns
ALE to Data Out Valid	t_{LD}		350	ns

Note: Timing is measured at V_L = 0.8 V and V_H = 2.2 V for both input and output
Load condition: C_L = 150 pF

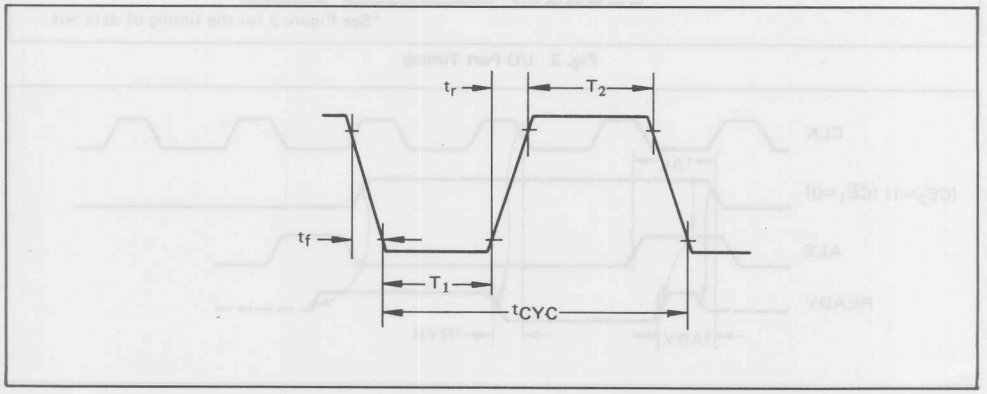


Fig. 1 Clock Signal for MSM83C55

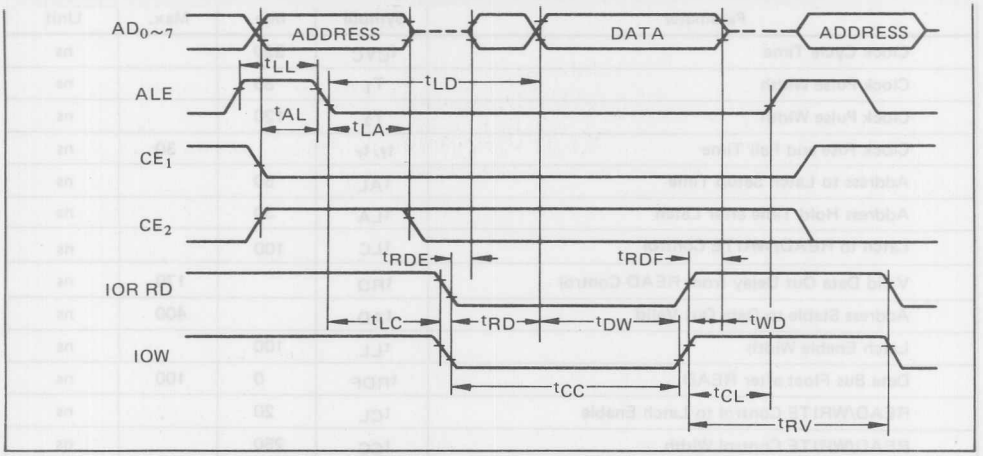


Fig. 2 Timing for ROM Reading and for I/O Reading and Writing

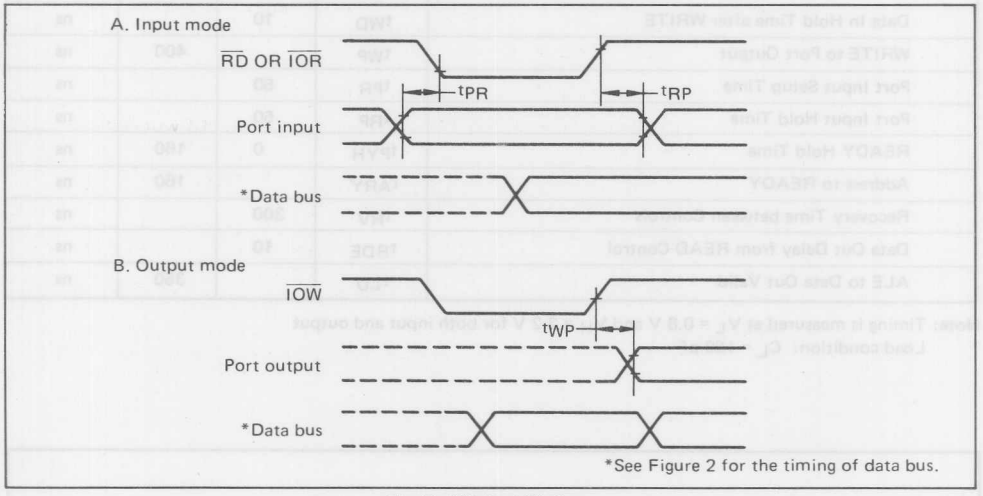


Fig. 3 I/O Port Timing

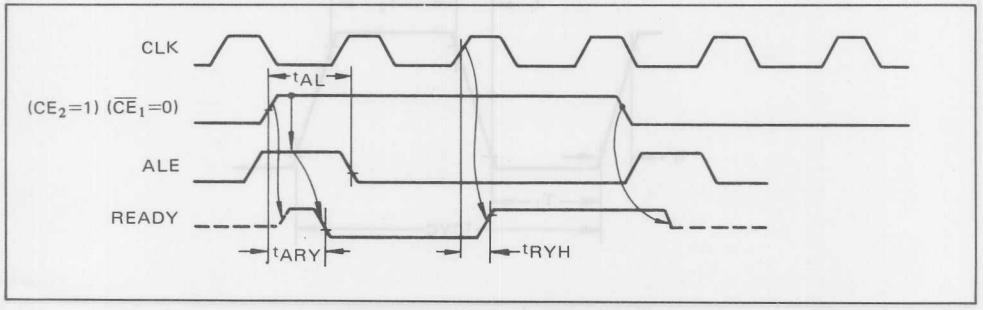


Fig. 4 Wait State Timing (READY = 0)

PIN DESCRIPTION

Pin symbol	Function
RESET	When this signal becomes high level, ports A and B becomes the input mode.
ALE	This pin is used to fetch the AD 0~7, A 8~10, IO/$\overline{M}$, $\overline{CE1}$, and CE2 signals to their respective latch circuits at the fall of the ALE (Address Latch Enable) signal.
$\overline{CE1}$, CE2	When $\overline{CE1}$ fetched to the latch circuit is high level or CE2 is low level, no read nor write operation is performed. The AD 0~7 and READY output signals are made into the floating status.
AD0~7	Three-stake bidirectional address/data bus. This bus fetch 8-bit address information to the latch circuit upon the fall of the ALE signal. When $\overline{CE1}$ in holding is low level and yet CE2 is high level, data is output from chip to bus if RD or IOR is low level and it is fetched from bus to chip if $\overline{IOW}$ is low level.
A8~10	These are high order bits of ROM address and have no relation to I/O operation.
IO/$\overline{M}$	When $\overline{RD}$ is low level, this pin selects I/O port if the IO/$\overline{M}$ in holding is high level or ROM if it is low level.
$\overline{RD}$	If the $\overline{RD}$ is low level, the memory data is output to AD 0~7 when the ROM cycle is selected, but the selected port data is output to the same when the I/O cycle is selected.
$\overline{IOR}$	The port data selected at low level is output to AD 0~7. When turned to the low level, the $\overline{IOR}$ becomes the same function as that when IO/$\overline{M}$ is turned to the high level and $\overline{RD}$ to the low level. When both $\overline{RD}$ and $\overline{IOR}$ become high level, the output of AD 0~7 is made into the floating state.
$\overline{IOW}$	At the low level, the AD 0~7 data is written to the selected port.
CLK	This signal is used to generate the READY signal for the generation of 1 wait cycle built in 83C55.
READY	This signal becomes low level when the ALE is high level and the $\overline{CE1}$ and CE2 are active. It becomes high level at the rise of CLK after the fall of the ALE.
PA0~7	These are universal I/O pins and the input/output is determined by the content of the data direction register. When writing data to port A, make the chip enable active and turn the $\overline{IOW}$ to low level after selecting AD 0, 1 to 0, 0. When reading it, turn the $\overline{IOR}$ to low level instead of $\overline{IOW}$ and IO/$\overline{M}$ to high level.
PB0~7	Same as the operation of PA0~7, excepting that AD0 is selected to 1 and AD1 to 0.
V_{CC}	+5 V power supply
GND	0 V

OPERATIONAL DESCRIPTION

ROM Block

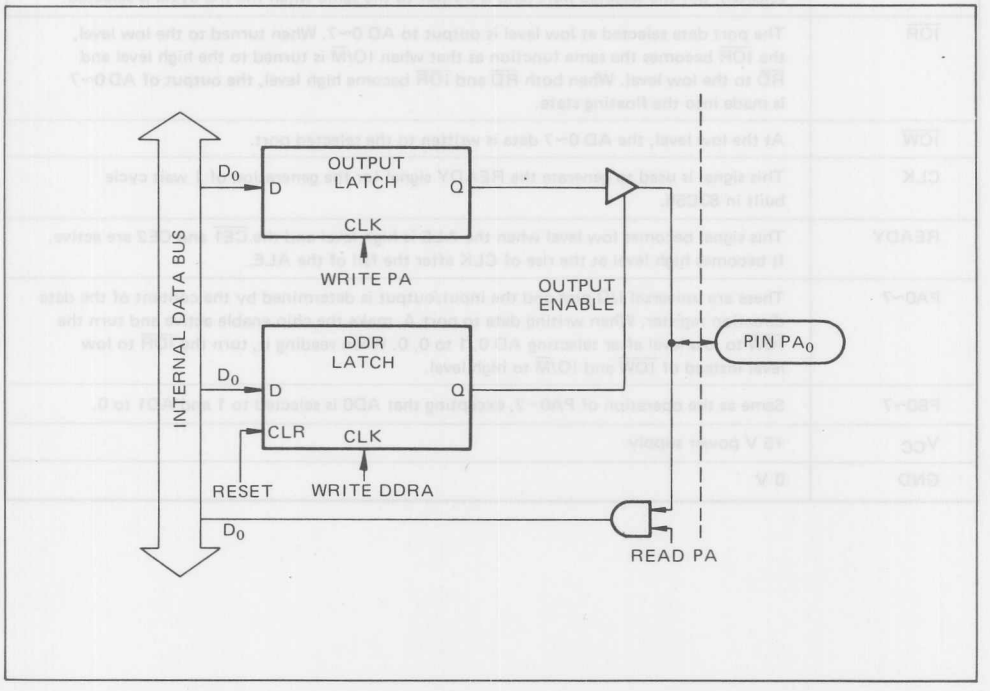
The ROM block in the chip is specified in address by the chip enable and 11-bit address. Upon the fall of the ALE signal, the address and chip enable are fetched in the address latch circuit. When the chip enable is active and IO/$\overline{M}$ is low level, 8-bit content of ROM at the address held in the address latch circuit is transmitted to the bus through the output buffer of AD 0~7 upon the fall of the $\overline{RD}$.

I/O Block

The I/O block in the chip is specified in address by the value of 2-bits of AD 0~1 and chip enable. Two 8-bit data direction registers (DDR) built in MSM83C55 are used to turn corresponding individual port pins to the input mode or output mode. It becomes the input mode when set to 0 and the output mode when set to 1. It is impossible to read the DDR from outside, however.

AD_1	AD_0	Selection
0	0	Port A
0	1	Port B
1	0	Port A data direction register (DDRA)
1	1	Port B data direction register (DDRB)

Upon the fall of $\overline{IOW}$ when the chip enable is active, the AD 0~7 data is written to the I/O port to be determined by the value of AD 0~1 in holding. During this operation, selected side I/O bits are all subject to its influence irrespective of the I/O status and IO/M status. The output level remains unchanged until the $\overline{IOW}$ returns to high level. The data can be read from ports when the chip enable in holding is active and IO/M is high level and yet the $\overline{RD}$ or $\overline{IOR}$ signal falls. In both input and output, the data on the selected side exists on the line of AD 0~7. The function of I/O ports and DDR (data direction register) is shown in the block diagram below:



Writing "0" to the DDR is equivalent to the RESET operation when the port output is made into the High impedance status and the input mode is specified. Note that the data can be written to the ports even if the output pin was already in the high impedance status (input mode) by the DDR. Likewise, it is also possible to read the data once set to those ports.